

Übungsaufgabe 1

zum Lernbuch »Grundlagen der PHP5-Programmierung«

Gästebuch (dateibasiert)

Im Folgenden werden Sie eine interaktive Web-Applikation zur Verwaltung eines Online-Gästebuchs erstellen.

Zieldefinition

Das Gästebuch soll folgende allgemeine Funktionen bereitstellen. Die genauen Beschreibungen folgen später in diesem Dokument:

- Alle Einträge des Gästebuchs werden auf der Startseite als Übersicht angezeigt.
- Zu jedem Gästebuch-Eintrag wird der Name des Autors und seine Webseite angezeigt.
- Alle Besucher sollen einen Gästebuch-Eintrag schreiben dürfen.
- Zur Datenhaltung der Gästebuch-Einträge soll eine Textdatei dienen.

Sie werden das Projekt in mehreren Schritten realisieren. Zu jedem Arbeitsschritt ist die Vorgehensweise beschrieben, die Ihnen zur Lösung der jeweiligen Anforderungen als Hilfestellung dienen soll. Die Angaben zur Vorgehensweise sind als Vorschläge zu verstehen.

Wichtig ist, dass Sie die Anforderungen jedes Arbeitsschritts konsequent erfüllen. Eigene Erweiterungen können Sie jederzeit einbauen, wenn Sie diese für sinnvoll erachten, aber die gestellten Anforderungen müssen mindestens erfüllt sein, da Sie nur für diese Punkte erhalten.

Technologien

Zur Realisierung der Applikation werden Sie folgende Technologien einsetzen:

- PHP5 (z.B. als Teil der Distribution XAMPP)
- HTML5

Folgende Technologien dürfen Sie verwenden, sind aber zur Realisierung des Projektes nicht zwingend notwendig:

- CSS
- HTML5-Formularvalidierungen
- JavaScript mit W3C-DOM

Vorgehensweise

Programmieren Sie die geforderten Funktionen des jeweiligen Arbeitsschrittes vollständig aus. Testen Sie anschließend die Funktionalität dieser und aller vorangegangenen Arbeitsschritte. Erst wenn Sie keine Probleme mehr finden, gehen Sie zum nächsten Arbeitsschritt über.

Wann immer Sie Funktionalität an mehreren Stellen des Gästebuchs benötigen (z.B. Auslesen der Einträge), lagern Sie den entsprechenden Code in Funktionen aus.

Halten Sie auch die gelernten Codestandards zum Beispiel zur Struktur von PHP-Dateien oder Namensgebung von Variablen/Funktionen ein.

Arbeitsschritt 1: Startseite erstellen

Anforderungen

- Auf der Startseite des Gästebuchs sehen die Besucher einen Bereich, der die Gästebuch-Einträge enthält und einen Link "Eintrag schreiben".

Vorgehensweise

1. Legen Sie eine Datei *index.php* an und planen Sie ein einfaches Layout für die geforderten Inhalte.
2. Ein Eintrag soll aus einer Überschrift, einem Textbereich und dem Satz "Verfasst von Autor am Datum um Uhrzeit" bestehen, wobei Autor, Datum und Uhrzeit später durch echte Daten ersetzt werden.
3. Fügen Sie drei Beispiel-Einträge als statischen Text in die Webseite ein. Diese werden in einem späteren Arbeitsschritt durch echte Einträge ersetzt.
4. Der Link "Eintrag schreiben" verweist auf eine Datei *eintrag_formular_anzeigen.php*. Legen Sie diese leer an.

Arbeitsschritt 2: Eintrag schreiben

Anforderungen

- Die Datei *eintrag_formular_anzeigen.php* zeigt ein Formular an, in das ein Besucher seinen Gästebuch-Eintrag schreiben kann.
- Das Formular enthält ein Eingabefeld für den Titel des Eintrags, ein mehrzeiliges Feld für den eigentlichen Eintrag und drei Felder, in die der Autor seinen Namen, seine E-Mail-Adresse und seine Homepage schreiben kann.
- Das aktuelle Datum des Eintrags soll ebenfalls gespeichert werden.
- Nachdem der Besucher einen Eintrag geschrieben und gespeichert hat, liest er die Nachricht "Vielen Dank für Ihren Eintrag".

Vorgehensweise

1. Die Seite *eintrag_formular_anzeigen.php* enthält ein Formular mit den Feldern `titel`, `inhalt`, `name`, `email` und `homepage`. Alle Felder, außer `inhalt` sind von Typ `input type="text"`, `inhalt` ist eine normale `textarea`.
2. Das Formular soll per POST an eine neue Seite *eintrag_speichern.php* versendet werden. Legen Sie diese leer an.
3. Die Datei *eintrag_speichern.php* legt die erhaltenen Formulardaten in ein mehrdimensionales Array `$eintraege` und speichert dieses serialisiert in die Datei *daten/eintraege.txt*. Die erste Ebene verwendet numerische Schlüssel. Die Schlüssel der inneren Arrays entsprechen den `name`-Attributen der Formularfelder.
4. Bei jedem Speichervorgang wird der neue Eintrag hinzugefügt, die bereits vorhandenen bleiben erhalten.
5. Bevor Sie den neuen Eintrag zum Array `$eintraege` ergänzen, soll das aktuelle Datum als Timestamp zum Eintrag hinzugefügt werden. Verwenden Sie hierfür den Array-Index `erstellungszeitpunkt`.
6. Nach dem Speichervorgang wird der Besucher auf eine neue Seite *eintrag_danke.php* weitergeleitet. Dies geschieht automatisch mit der PHP-Funktion `header()`.

Arbeitsschritt 3: Einträge anzeigen

Anforderungen

- Auf der Startseite werden nun die echten Einträge angezeigt, wobei der neueste oben in der Liste stehen soll.
- Wenn der Autor das Feld `homepage` ausgefüllt hat, soll sein Name in der Ausgabe mit dem angegebenen URL verlinkt sein. Hat er das Feld leergelassen, soll nur sein Name erscheinen.
- Das Feld `email` wurde zwar beim Speichern eines Eintrags erhoben, soll aber nirgendwo angezeigt werden.
- Wenn der Autor des Eintrags in der Textarea Zeilenumbrüche eingefügt hat, um einen Absatz zu verwenden, sollen diese Absätze auf der Startseite auch sichtbar sein.

Vorgehensweise

1. Ersetzen Sie die statischen Einträge durch die Inhalte der serialisierten Daten in `daten/eintraege.txt`. Behalten Sie dabei das bisherige Layout bei.
2. Wenn der Autor seine Homepage angegeben hat, also das Formularfeld `homepage` ausgefüllt hat, soll sein Name im Satz "Verfasst von Autor ..." mit dem angegebenen URL verlinkt sein.
3. Ersetzen Sie nun die Worte Datum und Uhrzeit im Satz "Verfasst von ..." durch das echte Datum und die Uhrzeit des Eintrags.
4. Ersetzen Sie die Zeilenumbrüche in der Textarea in der Ausgabe durch HTML `
`-Tags.

Arbeitsschritt 4: Sichere Ausgabe von Benutzereingaben

Anforderungen

- Alle dynamischen Inhalte, die aus Benutzereingaben stammen, dürfen keinen gefährlichen Code wie JavaScript oder HTML-Formatierungen enthalten.

Vorgehensweise

Sorgen Sie dafür, dass alle Benutzereingaben bei der Anzeige auf der Webseite (z.B. `index.php`) von HTML-Formatierungen und JavaScript befreit werden. Entscheiden Sie, ob die Funktion `strip_tags()`, `htmlspecialchars()` oder eine Kombination beider Funktionen für Ihr Gästebuch geeigneter ist.

Arbeitsschritt 5: Optionale Features

Betrachten Sie den folgenden Punkt als Vorschlag für eine bessere Usability der Webseite.

- Beim Eintrags-Formular sollen die Felder `titel`, `inhalt` und `name` Pflichtfelder sein. Nur wenn alle drei Felder ausgefüllt sind, soll das Formular abgeschickt und somit der Eintrag gespeichert werden.
- Außerdem soll das Feld `email` ein @-Zeichen enthalten müssen, sofern ein Wert angegeben wurde.

Optionale Features sind freiwillig und geben keine Bonuspunkte.

Vorgehensweise

1. Versehen Sie die Pflichtfelder mit dem HTML5-Attribut `required`.
2. Ändern Sie den Typ des Formularfelds `email` in `input type="email"`.

Übungsaufgabe 2

zum Lernbuch

»Grundlagen der PHP5-Programmierung«

Zitat des Tages

Im Folgenden werden Sie eine Web-Applikation erstellen, die Zitate anzeigt.

Zieldefinition

Die Applikation soll die folgenden allgemeinen Anforderungen erfüllen; die genauen Beschreibungen folgen später in diesem Dokument:

- Zur Datenhaltung soll eine MySQL-Datenbank dienen.
- Jeder Besucher soll Zitate in dieser Datenbank speichern können.
- Jeder Besucher soll sich auf der Startseite unserer Applikation ein Zitat anzeigen lassen können.
- Es wird immer nur genau ein zufälliges Zitat auf der Startseite angezeigt.
- Zu jedem Zitat wird der Name der zitierten Person angezeigt.
- Es wird in der Datenbank protokolliert, welches Zitat angezeigt wurde und "wer" es gesehen hat.
- Jeder eingeloggte Benutzer kann diese anonymisierten Protokoll-Daten auswerten.

Sie werden das Projekt in mehreren Schritten realisieren. Zu jedem Arbeitsschritt ist die Vorgehensweise beschrieben, die Ihnen zur Lösung der jeweiligen Anforderungen als Hilfestellung dienen soll. Die Angaben zur Vorgehensweise sind als Vorschläge zu verstehen.

Wichtig ist, dass Sie die Anforderungen jedes Arbeitsschritts konsequent erfüllen. Eigene Erweiterungen können Sie jederzeit einbauen, wenn Sie diese für sinnvoll erachten, aber die nachfolgend gestellten Anforderungen müssen **immer** erfüllt sein. Nur für diese erhalten Sie Punkte.

Technologien

Zur Realisierung der Applikation werden Sie folgende Technologien einsetzen:

- PHP5 (z.B. als Teil der Distribution XAMPP) mit PDO
- HTML5

Folgende Technologien dürfen Sie verwenden, sie sind aber zur Realisierung des Projektes nicht zwingend notwendig:

- CSS
- HTML5-Formularvalidierungen
- JavaScript

Vorgehensweise

Setzen Sie die Anforderungen des jeweiligen Arbeitsschrittes vollständig um. Testen Sie anschließend die Funktionalität dieser und aller vorangegangenen Arbeitsschritte. Erst wenn Sie keine Probleme mehr finden, gehen Sie zum nächsten Arbeitsschritt über.

Wann immer Sie eine Funktionalität an mehreren Stellen der Applikation benötigen (z.B. Formatierung von Datumsangaben), lagern Sie den entsprechenden Code in Funktionen aus. Mehrfach vorkommen-

der, identischer Code führt zu **Punktabzug**. Eine solche Auslagerung kann auch dann sinnvoll sein, wenn Ihr Code dadurch einfach nur lesbarer wird.

Halten Sie auch die gelernten **Codestandards** zum Beispiel zur Struktur von PHP-Dateien oder Namensgebung von Variablen/Funktionen ein.

Arbeitsschritt 1: Datenbankstruktur

Anforderungen

- Sie benötigen eine Datenbank-Tabelle `zitate` mit dem Primärschlüssel `id` und folgenden Spalten: `autor VARCHAR(255)`, `inhalt TEXT` und `erstellt_am DATETIME`.
- Sie benötigen außerdem eine Datei `reset.php`, die diese Tabelle und ein paar Beispieldaten anlegt.

Vorgehensweise

1. Legen Sie die Datei `reset.php` an, die zunächst nur die benötigte Tabelle erstellt. Verwenden Sie hierfür die schon bekannte `datenbank.inc.php`. Konfigurieren Sie die Datenbankverbindung und rufen Sie dann die `reset.php` im Browser auf.
2. Verwenden Sie nun Prepared Statements, um ein paar Zitate in dieser Tabelle abzulegen.
3. Ein Zitat könnte beispielsweise *“Erfolg hat drei Buchstaben: TUN!” (Johann Wolfgang von Goethe)* sein.
4. Erfassen Sie bei jedem Zitat, wann es in der Datenbank hinzugefügt wurde. **Tipp:** Dies geht besonders einfach mit der MySQL-Funktion `NOW()` direkt im SQL-Code.
5. Nehmen Sie abschließend eine Header-Umleitung zur Startseite `index.php` vor, die Sie im nächsten Schritt erstellen werden.

Arbeitsschritt 2: Startseite erstellen

Anforderungen

- Auf der Startseite des Gästebuchs sehen alle Besucher einen Bereich, der ein zufälliges Zitat enthält, und einen Link “Zitat anlegen”.
- Der Link soll zukünftig Teil eines Menüs sein. Setzen Sie dieses Menü also direkt im HTML-Code um, auch wenn es derzeit nur einen Link enthält.
- Wenn der Autor eines Eintrags im eigentlichen Zitattext Zeilenumbrüche eingefügt hat, um einen Absatz zu erzeugen, sollen diese Absätze auch auf der Startseite sichtbar sein.

Vorgehensweise

1. Legen Sie eine Datei `index.php` an und setzen Sie ein einfaches Layout/Design für die geforderten Inhalte um.
2. Ein Eintrag soll bei der Anzeige aus der Spalte `inhalt` und der Spalte `autor` bestehen. **Tipp:** Verwenden Sie die Funktion `nl2br()`, um Zeilenumbrüche bei der Zitatausgabe sichtbar zu machen.
3. Außerdem soll angezeigt werden, wann das Zitat hinzugefügt wurde (Spalte `erstellt_am`). Verwenden Sie hierfür ein formatiertes Datum mit Uhrzeit.
4. **Tipp:** Beim Auslesen eines zufälligen Zitats bietet sich die Verwendung von `ORDER BY RAND()` und `LIMIT 1` direkt im SQL-Code an.
5. Der Link “Zitat anlegen” verweist auf eine Datei `anlegen.php`. Legen Sie diese leer an.

Arbeitsschritt 3: Zitat anlegen

Anforderungen

- Die Datei *anlegen.php* zeigt ein Formular an, in das ein Besucher ein Zitat eintragen kann.
- Das Formular enthält genau zwei Felder: Ein mehrzeiliges Feld für das eigentliche Zitat und ein Feld, in das man den Namen der zitierten Person schreiben kann. Beide Felder sind entsprechend beschriftet.
- Das Erstellungsdatum des Eintrags soll ebenfalls gespeichert werden.

Vorgehensweise

1. Die Seite *anlegen.php* enthält ein Formular mit den Feldern `inhalt` und `autor`, wobei `inhalt` eine Textarea sein soll.
2. Das Formular soll per POST an die aktuelle Seite versendet werden.
3. Die Datei *anlegen.php* speichert die erhaltenen Formulardaten inklusive des aktuellen Datums in der Tabelle `zitate`.
4. Bei jedem Speichervorgang wird der neue Eintrag hinzugefügt, die bereits vorhandenen bleiben erhalten.
5. Nach dem Speichervorgang wird der Besucher auf die Startseite *index.php* weitergeleitet.

Arbeitsschritt 4: Sichere Ausgabe von Benutzereingaben

Anforderungen

- Alle dynamischen Inhalte, die aus Benutzereingaben stammen, dürfen keinen gefährlichen Code wie JavaScript oder HTML-Formatierungen enthalten. Filtern Sie diese Daten deshalb bei der Ausgabe.
- **Achtung:** Dies gilt auch für alle in den nachfolgenden Arbeitsschritten gespeicherten Benutzereingaben!

Vorgehensweise

Sorgen Sie dafür, dass alle Benutzereingaben bei der Anzeige auf der Webseite (z.B. *index.php*) von HTML-Formatierungen und JavaScript befreit werden. Entscheiden Sie, ob die Funktion `htmlspecialchars()`, `strip_tags()` oder eine Kombination beider Funktionen für die verschiedenen Werte geeigneter ist.

Arbeitsschritt 5: Logging

Anforderungen

- Der Betreiber der von Ihnen erstellten Applikation möchte gerne etwas mehr über seine Besucher erfahren und benötigt deswegen ein Logging von bestimmten Daten.
- Sie benötigen eine zweite Datenbank-Tabelle `log` mit dem Primärschlüssel `id`, dem Fremdschlüssel `zitat_id` und folgenden weiteren Spalten: `ip VARCHAR(32)`, `browser VARCHAR(255)`, `sprache VARCHAR(5)` und `erstellt_am DATETIME`.
- Beim Aufruf der Startseite (und nur dann) soll für den aktuellen Benutzer und Zeitpunkt ein Eintrag in dieser Tabelle erzeugt werden.
- Die Werte für die benutzerspezifischen Spalten sollen aus `$_SERVER` stammen, und zwar, um genau zu sein, aus: `REMOTE_ADDR`, `HTTP_USER_AGENT` und `HTTP_ACCEPT_LANGUAGE`. Da diese Werte im Browser manipulierbar sind, zählen sie übrigens auch zu den Benutzereingaben!

Vorgehensweise

1. Erweitern Sie die *reset.php*, damit diese fortan auch die zweite Tabelle erzeugt, und rufen Sie sie anschließend im Browser auf.
2. Erweitern Sie die *index.php* zwischen dem Auslesen des zufälligen Zitats und dessen Anzeige. Hier soll fortan ein Eintrag in der Tabelle `log` erfolgen.
3. Erzeugen Sie hierfür zunächst ein assoziatives Array `$log`, welches die drei Werte aus `$_SERVER` enthält und diese mittels der korrekten Schlüssel `ip`, `browser` und `sprache` den drei Spalten zuordnet.
4. Kürzen Sie nun den Wert aus `$_SERVER['HTTP_ACCEPT_LANGUAGE']` im Array. Uns interessiert nur die erste Angabe, also die bevorzugte Sprache.
5. Ergänzen Sie abschließend den Fremdschlüssel `zitat_id` im Array und speichern Sie den kompletten Datensatz in der Tabelle `log`.

Arbeitsschritt 6: Auswertungen

Anforderungen

- Im Menü soll es nun einen weiteren Link "Auswertungen" geben.
- Beim Aufruf dieses Links soll eine Anzeige aller Datumswerte erfolgen, für die es Log-Einträge gibt.
- Es soll wirklich nur das jeweilige Datum (ohne Uhrzeit) und jedes Datum nur **ein einziges Mal** angezeigt werden.
- Das neueste Datum soll ganz oben ausgegeben werden.
- Jedes Datum soll mit einer entsprechenden Detailseite verlinkt sein.

Vorgehensweise

1. Legen Sie eine Datei *auswertungen.php* an und verlinken Sie diese im Menü.
2. Lesen Sie in dieser Datei nur die Datumswerte aller Log-Einträge in der korrekten Reihenfolge aus.
Tipp: Dies geht beispielsweise mit `SELECT DATE(erstellt_am) FROM log`.
3. Wir lesen in diesem speziellen Fall nur den Wert einer einzigen Spalte der Tabelle aus. Trotzdem wird uns die Methode `PDOStatement#fetchAll()` ein mehrdimensionales Ergebnis-Array zurückliefern.
Tipp: Beim Aufruf der Methode können Sie dies mit dem Parameter `PDO::FETCH_COLUMN` ändern.
4. Lesen Sie keine doppelten Datumswerte aus.
Tipp: Hierbei könnte das Schlüsselwort `DISTINCT` in der SQL-Anweisung sehr nützlich sein. Alternativ könnte man auch `array_unique()` auf das Ergebnis-Array anwenden.
5. Zeigen Sie nun die Datumswerte mittels einer Schleife an.
6. Jeder Datumswert verweist auf eine Datei *details.php*, d.h. jeder Wert wird als Link dargestellt. Legen Sie diese neue Datei leer an, wir werden sie dann im nächsten Schritt befüllen.
7. Übergeben Sie als URL-Parameter an diese Datei, welches Datum angeklickt wurde.

Arbeitsschritt 7: Auswertungsdetails

Anforderungen

- Durch Anklicken eines Datums in der Datumsauswahl wird eine Seite aufgerufen, die alle Log-Einträge für diesen Tag anzeigt. Es reicht hierbei, wenn für das Zitat lediglich die ID ausgegeben wird.

- Unter dieser Anzeige möchte der Applikations-Betreiber sehen können, wie viele Besucher es insgesamt gab, wie hoch der Anteil von Windows- und Mac OS-Nutzern war, wie viele Besucher als bevorzugte Sprache deutsch hatten und welche Zitate wie oft angezeigt wurden.
- Wir benötigen also drei Anzeigebereiche: Für den Log-Ausschnitt des gewählten Tages, für die absolute/prozentuale Auswertung der Besucherzahlen dieses Tages und für die Anzeigezahlen der Zitate an diesem Tag.

Vorgehensweise

1. Ermitteln Sie in der *details.php* alle Log-Einträge des ausgewählten Tages und bestimmen Sie deren Anzahl.
2. Ermitteln Sie nun, wie viele dieser Besucher Windows- bzw. Mac OS-Nutzer waren. Die Browser-Angabe sollte im ersten Fall `Windows` und im zweiten Fall `Mac OS` enthalten.
3. Ermitteln Sie danach, wie viele dieser Besucher deutsch (Wert `de-DE` bzw. beim Firefox nur `de`) als bevorzugte Sprache eingestellt hatten.
4. Ermitteln Sie abschließend alle Zitat-IDs des Tages und wie oft diese vorkamen.
Tipp: Die Funktion `array_count_values()` könnte hierbei eventuell hilfreich sein. Recherchieren Sie deren Funktionsweise.
5. Zeigen Sie alle diese Werte in der *details.php* an.

Arbeitsschritt 8: Login/Logout

Anforderungen

- Nur eingeloggte Benutzer können die Auswertungen ansehen. Ein Zugriff für nicht eingeloggte Benutzer darf nicht möglich sein!
- Die Benutzerdaten sollen direkt im Code und nicht in der Datenbank hinterlegt sein. Eine Registrierung für neue Benutzer gibt es nicht!
- Es sollen mindestens zwei Benutzer (mit unterschiedlichen Benutzernamen und Kennwörtern) hinterlegt sein.
- Das Menü soll zudem bei eingeloggten Benutzern eine Möglichkeit zum Logout anbieten.
- Die Seite zur Anzeige der Auswertung eines bestimmten Tages soll nur aufrufbar sein, wenn der Benutzer eingeloggt ist. Ist er dies nicht, soll der Besucher zur Startseite weitergeleitet werden.

Vorgehensweise

1. Zeigen Sie in der *auswertungen.php* ein Login-Formular an, sofern der Benutzer noch nicht eingeloggt ist.
2. Hinterlegen Sie die korrekten Benutzerdaten im Code (z.B. in der *konfiguration.inc.php*).
3. Prüfen Sie, ob die im Formular eingegebenen Benutzerdaten korrekt sind. Geben Sie bei einem fehlerhaften Login eine entsprechende Rückmeldung.
4. War der Login erfolgreich, so merken Sie sich dies in der Session.
5. Erst wenn in der Session ein korrekter Login vermerkt wurde, soll wieder die verlinkte Datumsübersicht in der *auswertungen.php* sichtbar sein. Auch die hierfür nötigen Daten muss man übrigens nur für eingeloggte Benutzer aus der Datenbank auslesen.
6. Schützen Sie die *details.php* gegen den Aufruf durch unberechtigte Besucher.

Arbeitsschritt 9: Anonymisierung

Anforderungen

- Die Speicherung personenbezogener Daten unterliegt in Deutschland strengen Vorgaben. Die IP-Adresse des Besuchers ist eine solche Information.
- Damit die Speicherung der IP-Adresse zulässig ist, wünscht der Betreiber eine Anonymisierung der Information.
- **Hinweis:** Wenn hierdurch einzelne Besucher nicht mehr zwingend unterscheidbar sind, so ist dies für die Umsetzung der Applikation nicht relevant.

Vorgehensweise

1. Überarbeiten Sie das assoziative Array `$log`. Der Wert von `$_SERVER['REMOTE_ADDR']` muss nun vor der Speicherung anonymisiert werden.
2. Erstellen Sie eine Funktion `anonymisiereIp()`, die diese Anonymisierung für den Parameter `$ip` vornimmt. **Tipp:** Ermitteln Sie zunächst mit `md5($ip)` einen sogenannten Hash `$hash` für die IP-Adresse. Dieser hat eine Länge von 32 Zeichen. Nun ist die IP-Adresse nicht mehr direkt erkennbar, aber immer noch nicht anonymisiert. Dies erreichen wir erst mit `substr($hash, 0, 15)`, d.h. wir behalten weniger als die Hälfte der Zeichen des Hashs.

Arbeitsschritt 10: Optionale Features

Optionale Features sind freiwillig und geben **keine** Bonuspunkte. Betrachten Sie den folgenden Punkt als Vorschlag für eine bessere Usability der Webseite.

- Die Besucher verwenden Browser-Reloads, um mehr als ein Zitat pro Tag sehen zu können. Hierdurch werden die Logs verfälscht, was nicht erwünscht ist.
- Beim Loggen der Besucher auf der Startseite soll deshalb jeder eindeutige Besucher nur ein einziges Mal pro Tag erfasst werden, und zwar mit dem ersten gesehenen Zitat.
- Ein Besucher ist eindeutig, sofern die anonymisierte IP-Adresse, der Browser und die bevorzugte Sprache identisch sind.

Vorgehensweise

1. Prüfen Sie in der `index.php` vor dem Loggen, ob es einen Besucher mit identischen Werten in den Spalten `ip`, `browser` und `sprache` an diesem Tag schon gab.
2. Gab es keinen solchen eindeutigen Besucher, so soll ein Logging mit der aktuellen Zitat-ID erfolgen. Gab es den eindeutigen Besucher hingegen heute schon, so soll er nicht erneut geloggt werden.

Weitere optionale Features sind denkbar. Beispielsweise wäre es sinnvoll, wenn neue Zitate (vor der ersten Anzeige) durch einen eingeloggten Benutzer freigeschaltet werden müssten.