

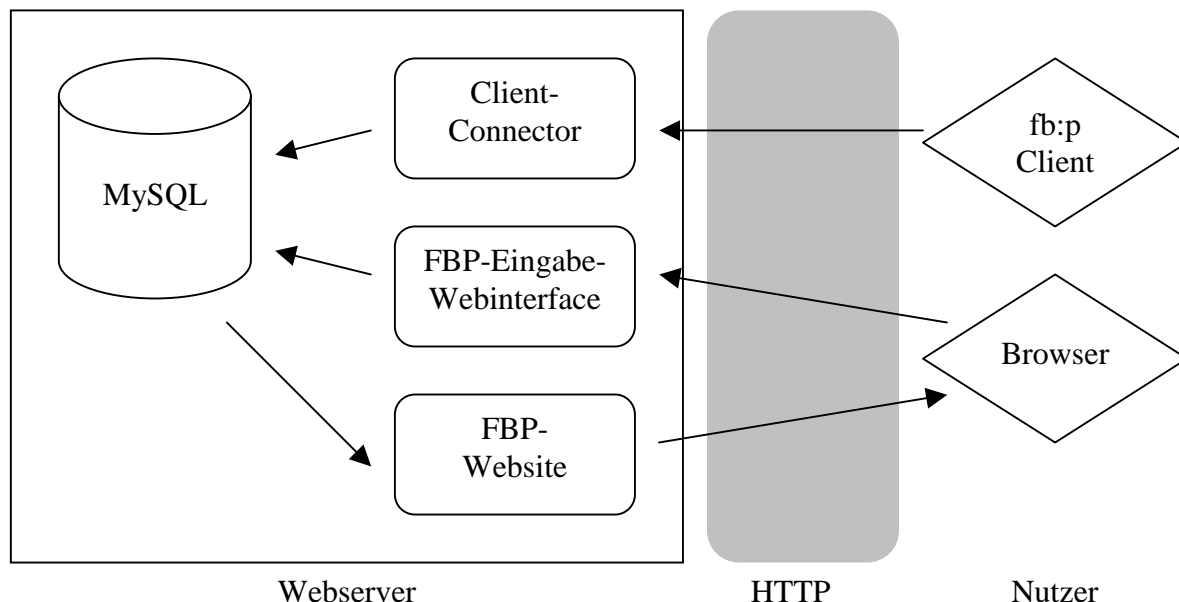
Dokumentation der Server-API-Funktionen

I. Grundkonzept

fb:porticula ist ein erweitertes NoPaste-System, das von FreeBasic-Portal.de zur Verfügung gestellt wird. Ein NoPaste-System dient grob gesagt der Speicherung von Code-Schnipseln im Internet, damit man sie z.B. in Foren, Chats usw. zeigen kann. Eigener Webspace ist dazu also nicht nötig. fb:porticula kann Quellcodes aber nicht nur speichern und bei Bedarf anzeigen, sondern auch automatisch einem sog. Syntax-Highlighting unterziehen. Dabei werden Schlüsselwörter und andere Sprachelemente in verschiedenen Formatierungen dargestellt, sodass der Quellcode leicht zu überschauen wird.

Das Besondere an fb:porticula ist, dass man nicht nur über die Website (eine HTML Eingabemaske) Quellcodes posten kann, sondern dass man mit dem eigens von uns entwickelten fb:porticula Clientprogramm für Windows mit einer Tastenkombination Quellcodes aus der Zwischenablage automatisch auf die Website hochladen kann.

Serverseitig besteht fb:porticula aus php-Skripten und einer MySQL-Datenbank. Die php-Skripte sorgen auf der einen Seite für die formatierte Ausgabe der in der Datenbank befindlichen Quelltexte und auf der anderen Seite für die Annahme und Speicherung neuer Quelltexte. Sowohl über ein in die FBP-Website integriertes Webinterface als auch über die erwähnte Clientsoftware können Quellcodes in fb:porticula abgelegt werden. Die Verbindung zwischen Speicher-PHP-Skript und Clientsoftware erfolgt über genormte http-Aufrufe der servergespeicherten PHP-Skripte.



Der neue fb:porticula Client wird in FreeBASIC mit GTK+ entwickelt, sodass er auf MS Windows und Linux gleichermaßen zur Verfügung steht. Die alte Clientsoftware, die in VisualBasic für Windows entwickelt wurde, wird durch die neue Clientsoftware abgelöst. Abwärtskompatibilität zur alten Clientsoftware ist nicht vorgesehen.

II. http-Aufrufe

Die im Folgenden beschriebenen Aufrufe sind relativ zur Basisadresse `http://www.freebasic-portal.de/porticula.php?`

Alle Parameter, die an das serverseitige Script zu übergeben sind, werden als GET-Variablen über http-Anfragen übermittelt.

Des Weiteren ist zu beachten, dass das Passwort immer als MD5-Hash übergeben werden muss. Der Benutzername muss wie alle anderen via Query String übermittelten Inhalte escaped werden. Dies ist von besonderer Bedeutung, wenn beispielsweise Leerzeichen oder Umlaute in den Strings enthalten sind. Wenn bei der Verarbeitung des Befehls Fehler auftreten, wird stets die Zeichenfolge `ERROR:` gefolgt von einer kurzen Fehlerbeschreibung zurückgegeben.

Versions- informationen usw. anzeigen	<code>action=info</code> Gibt Informationen zum Connector, der aktuellen Clientprogrammversion und zur Verbindung zum Client im Textformat zurück.
Beiträge eines Benutzers auflisten	<code>action=list&username=<??></code> Gibt eine Liste aller dem Autor zuzuordnenden Porticula-Einträge im CSV-Format (Text) zurück. Das Trennzeichen dabei ist ein Semikolon, das, falls es in einem String vorkommt, serverseitig durch <code><semikolon></code> ersetzt wird. Die Felder sind nicht von Anführungszeichen umschlossen, das Datensatzabschlusszeichen ist <code>CHR(10)</code> , was dem Zeilenumbruch auf UNIX-Systemen entspricht. Format der Datensätze: <code>{DocID};{Datum/Zeit};{Titel};{Dateigröße}<CHR(10)></code>
Alle Datensätze auflisten	<code>action=listall&n=<??>&offset=<??></code> Gibt eine Liste aller Datensätze im CSV-Format (Text) zurück, die stets absteigend nach dem Erstellungsdatum sortiert ist. Das Format der Rückgabe ist nahezu identisch zu der von <code>list</code> (s.o.), enthält allerdings eine 5. Spalte, die den Benutzernamen des Autors angibt. Der Parameter <code>n</code> legt fest, wie viele Einträge ausgegeben werden sollen. <code>n</code> darf entweder eine Ganzzahl größer/gleich 1 oder <code>all</code> sein. Mit dem <u>optionalen</u> Parameter <code>offset</code> (positive Ganzzahl) kann angegeben werden, wie viele Datensätze übersprungen werden sollen, was nützlich ist, wenn die Einträge nicht komplett sondern seitenweise ausgegeben werden sollen.
Anzahl aller Datensätze ausgeben	<code>action=count</code> Gibt die Anzahl aller NoPaste-Datensätze aller Benutzer zurück.

URL eines Datensatzes mittels ID ermitteln	<code>action=id2url&id=<??></code> Gibt die vollständige Adresse (inkl. http://) zum Datensatz mit der übergebenen ID oder im Fehlerfall ERROR zurück.
Datensatz herunterladen	<code>action=download&ID=<??></code> Gibt einen datenbankgespeicherten Quelltext sowie Details dazu zurück. Kann auch alternativ zu <code>readcode</code> verwendet werden, gibt aber anders als <code>readcode</code> auch den Usernamen des Autors zurück. Format der Datenausgabe: {DocID}<CHR(10)> {Datum/Zeit}<CHR(10)> {Username}<CHR(10)> {Titel}<CHR(10)> {Quelltext}
Eigenen Beitrag mit bestimmter ID löschen	<code>action=delete&ID=<??>&username=<??>&pwhash=<??></code> Löscht einen Code-Eintrag aus der Porticula-Datenbank. Es können nur Beiträge gelöscht werden, die von dem Benutzeraccount, dessen Zugangsdaten übergeben werden, erstellt worden sind. Gibt SUCCESS bei erfolgreichem Löschen oder ERROR-Code (s.o.) zurück.
Quelltext aus Datenbank abrufen, um ihn zu bearbeiten	<code>action=readcode&ID=<??>&username=<??>&pwhash=<??></code> Gibt einen datenbankgespeicherten Quelltext sowie Details dazu zurück. Es kann über den Aufruf nur auf Quelltexte zugegriffen werden, die von dem Benutzeraccount, dessen Zugangsdaten übergeben werden, erstellt worden sind. Format der Datenausgabe: {DocID}<CHR(10)> {Datum/Zeit}<CHR(10)> {Titel}<CHR(10)> {Quelltext}
Geänderten Datensatztitel und Quelltext in Datenbank übernehmen	<code>action=savecode&ID=<??>&username=<??>&pwhash=<??></code> Neben dem o.g. Query-String mit den enthaltenen GET-Variablen müssen im POST-Teil der http-Anfrage Werte für Titel und Quelltext übergeben werden. Die aktuell in der Datenbank befindlichen Vorgabewerte können über <code>action=readcode</code> (s.o.) ausgelesen und in eine entsprechende Eingabeoberfläche geladen werden. Es kann über den Aufruf nur auf Quelltexte zugegriffen werden, die von dem Benutzeraccount, dessen Zugangsdaten übergeben werden, erstellt worden sind. Der Server gibt im Erfolgsfall SUCCESS oder eine ERROR-Nachricht bei Fehlschlag zurück.

**Neuen
Datensatz
erstellen**

`action=newcode&username=<??>&pwhash=<??>`

Neben dem o.g. Query-String mit den enthaltenen GET-Variablen müssen im POST-Teil der http-Anfrage Werte für `Titel` und `Quelltext` übergeben werden. Der Server gibt im Erfolgsfall `SUCCESS` gefolgt von einem Leerzeichen und der Datensatz-ID des neuen Eintrags zurück.
Beispiel: `SUCCESS 123`

**Zugangsdaten
überprüfen**

`action=check&username=<??>&pwhash=<??>`

Eingegebene Zugangsdaten können auf Richtigkeit überprüft werden. Gibt `SUCCESS` zurück, wenn die Kombination von `Benutzername` und `Passworthash` in der Datenbank gefunden wurde. Nach einer serverseitig festgelegten Anzahl von fehlgeschlagenen Authentifizierungsversuchen wird die Client-IP-Adresse für einen bestimmten Zeitraum gesperrt und kann nicht mehr mit dem Porticula-Connector in Verbindung treten. Ist die IP-Adresse geblockt, wird – egal welche Anfrage an den Server geschickt wird – `BLOCKED` zurückgegeben.

III. Versionsüberprüfung / Updates

Beim Programmstart der Clientsoftware sollte mit der Anfrage `action=info` überprüft werden, ob die Clientsoftware noch auf dem neusten Stand ist. Die Clientsoftware sollte des Weiteren ihre Version über die Angabe des `HTTP_USER_AGENT`s dem Server mitteilen. Wie im Falle neuer Programmversionen auf dem Server vorzugehen ist, ist noch zu evaluieren. Optimal wäre der automatische Download und die automatische Einrichtung der neuen Programmdateien, wie dies beim `fb:porticula Client 1.0` geschehen ist. Der Bezug der neuen Programmdateien würde in jedem Fall über `http` vom `freebasic-portal.de` Server erfolgen.