

## Instalace systému

Instalace systému je velmi jednoduchá, pokud jsou splněny nefunkční požadavky na systém, na kterém má běžet aplikační serverová logika a databáze. Aplikaci stačí nakopírovat do složky přístupné přes webové rozhraní. Databázi stačí na databázovém serveru vytvořit jako prázdnou bez tabulek, stačí, aby administrátor sdělil údaje pro přihlášení a název databáze. Aplikace se postará o vytvoření schéma databáze a vloží do databáze uživatele s nejvyššími právy editor, tento uživatel je přístupný pod jménem ,admin‘ a heslem ,admin‘, velmi se doporučuje toto nastavení co nejdříve změnit, protože se jedná o známé často využívané pojmenování k přístupu k účtům.

## Nastavení databáze a aplikace

Aplikace je vytvořena tak, aby umožňovala jednoduché nastavení přístupových údajů k databázi pomocí konfiguračního souboru, kde je vše přehledně zapsáno na jediném místě. Tento soubor je zabezpečen, aby se k němu nedalo dostat přes webový prohlížeč, protože obsahuje důležité ověřovací údaje k přístupu k serveru. Obsah konfiguračního souboru je kromě části ,common‘, kde jsou některá běžná nastavení aplikace rozdělen na části ,production‘ pro nastavení v produkčním prostředí a ,development‘ pro nastavení prostředí pro testování. Nastavení databáze funguje na stejném principu pro obě sekce, nyní si popíšeme jeho strukturu včetně názvů parameterů:

- `database.count = 1` – obsahuje počet databází v nastavení, podle tohoto počtu aplikace prochází dál nastavení a hledá konfiguraci jednotlivých databází s názvem `databaseX`, kde X nabývá hodnot 1 až po počet databází
- `database1.driver = mysql` – ovladač databázového systému databázové vrstvy Dibi
- `database1.host = localhost` – jmenná adresa nebo IP serveru s databází
- `database1.charset = utf8` – znaková sada pro komunikaci
- `database1.lazy = true` – zda se má spojení vytvořit okamžitě při volání funkce pro spojení s databází (hodnota `false`) nebo takzvané líné spojení, kdy se má spojení vytvořit až při prvním dotazu pro databázi
- `database1.database = database-name` – jméno databáze na serveru
- `database1.username`, `database1.password` – přihlašovací údaje

- `database1.name = ,Databáze organizace‘` – název pro použití v aplikaci pro označení databáze, nemá žádnou souvislost s připojením k databázi

Takto můžeme velmi jednoduše nastavit aplikaci několik databází, ke kterým se může připojit, tudíž jedna aplikace může sloužit více organizacím, kdy každá bude mít vlastní databázi. Navíc není nutně vyžadován databázový server MySQL protože databázová vrstva využitá v aplikaci má možnost se připojit pomocí svých ovladačů i k jiným typům databází. Jedinou nevýhodou v tuto chvíli je, že aplikace v této verzi nepodporuje vytváření záloh pro jiný typ připojení přes PHP než je `mysqli` a to z důvodů odlišného vytváření zálohy pro jednotlivé databázové systémy. Požadavkem na tuto aplikaci bylo právě využití MySQL, funkcionality na jiných databázích je tedy navíc a chybějící zálohování je tedy jedna z možností jak aplikaci dále v budoucnu rozšířit.

Podporované databázové systémy knihovnou Dibi:

- MySQL v obou variantách pro PHP tedy klasické a objektové `MySQLi`,
- PostgreSQL,
- SQLite,
- Firebird,
- ODBC,
- PDO,
- Oracle,
- experimentálně MS SQL včetně odlišného ovladače pro verzi 2005.

Nastavení aplikace nevyžaduje žádná speciální nastavení kromě přístupu k databázi. Pokud webové úložiště aplikace bude umožňovat přístup do složek, které nebudou viditelné přes webový server, přesunout je a upravit cestu k nim v konfiguračním souboru. Obvykle kvalitní webový hosting nabízí prostor, který obsahuje zvolenou složku například s názvem `www`, která je přístupná právě přes webový server pod sjednanou adresou. Mimo tuto složku si tedy můžeme přesunout složky s aplikací, knihovnami a adresářem pro dočasné soubory, což vede ke zvýšení bezpečnosti uložení tajných dat o aplikaci. V případě změny struktury adresářů tedy stačí upravit soubor v kořenovém adresáři webu a nastavit v něm příslušné cesty. Některá webová úložiště nenabízí tuto možnost a proto je aplikace v základu koncipována tak, že

jednotlivé tajné složky jsou chráněny nastavením vlastností přístupu přes webový server, což je provedeno skrze soubory v nich, které si server čte při vstupu do nich.

## **Údržba systému**

Systém z technického hlediska nevyžaduje speciální požadavky na jeho udržování v chodu. Pokud hardware i software, na kterém aplikace běží nezpůsobí nějakou chybu v systému, což může způsobit i aplikace samotným při některém neočekávaném a neošetřeném stavu, tak by nemělo být zapotřebí se o aplikaci jakkoliv starat. V případě neošetřené události nebo výjimky si aplikace vytváří zápis do logovacího souboru s podrobným popisem stavu aplikace a zdrojem události. Dalším z problémů, které mohou vzniknout, jsou problémy s databází. Může dojít například k jejímu naplnění, i když v tuto chvíli je její struktura koncipována tak, aby při jejím očekávaném běžném využití nedošlo k jejímu naplnění. Vzhledem k malému množství záznamů z pohledu databázového systému by aplikace mohla sloužit i několik desítek let, takže by měla bez problémů sloužit po dobu používání tohoto informačního systému. Pokud by ovšem tato nepředpokládaná situace nastala, bylo by potřeba změnit některé vlastnosti schématu databáze, tak aby umožňovala vložení více záznamů. Případně může dojít k poklesu výkonu databázového systému, potom by tedy bylo nutné buď pozměnit schéma databáze, nebo zvážit změnu systému, problémů může být několik a to nejen na straně databáze a jejího systému. Obdobný problém se týká přílišného vytížení aplikačního serveru, kdy by se měla monitorovat doba vykonávání skriptů a pokud výkon klesne pod přijatelnou úroveň, tak by měla být provedena příslušná opatření.

## **Zálohování**

Informační systém využívá vícero datových úložišť. Servery by měli být automaticky zálohovány, ale pokud se nechceme spoléhat na tuto formu, můžeme manuálně zálohovat také. Jedno úložiště je na webovém serveru pro ukládání příloh, je společné pro všechny nastavené databáze. Z informačního systému nejde jednoduše zálohovat, protože na práci s větším objemem dat není PHP navrženo, proto se nabízí možnost přihlášení se na server a ručního stažení příslušné složky. U databáze je zálohování velmi jednoduché, stačí se přihlásit do systému s editorskými právy a zvolit odkaz „Záloha databáze“ a uložit soubor. Případnou obnovu zálohy může vytvořit administrátor databáze nebo ji lze nakopírovat místo inicializační kostry prázdné databáze, která se využívá při prvotním připojení k prázdné databázi bez struktury.