# Fra design til kode – men hvad med databasen?

|  |  |  |
| --- | --- | --- |
| OK så nu har vi bygget en simpel model af klasser og et design klasse diagram som et proof-of-concept på modelniveau. På modelniveau lever data om objekterne i hukommelsen af maskinen, intet skrives i filer, database, sessionobjects hverken lokalt på maskinen eller i skyen for den sags skyld.  **Næste skridt er at bringe vores database i spil. Hvordan kan vi gøre det?** |  |  |

Det kan vi som med så meget andet gøre på mange forskellige måder. I 2 screencasts gennemgår vi nogle webartikler, hvor forfatteren med kode eksempel der anvender database forklarer om mvc, eller giver bud på praktisk implementering af et eller andet specifikt system som.

* Anvender **database** med CRUD (CREATE READ UPDATE DELETE)
* har forskellige bud på simpel **Model-View-Control** implementering

|  |  |
| --- | --- |
| **Client server intro**  (28 minutter)  **Website:**  PHP MVC With Database (Very simple beginner)  <https://code-boxx.com/simple-php-mvc-example/#sec-download>  **Screencast**  <https://www.studietube.dk/video/7478230/e1abe4cee3680d0158d25c303fb4a3af>  **Git**  [**https://github.com/kimo1ucl/Database-Connectivity-Examples**](https://github.com/kimo1ucl/Database-Connectivity-Examples)  Folder name: mvc-php-mysql | **Indhold:**  Kortere kan det næsten ikke blive. 3 simple MVC klasser hvor model implementerer databaseadgangen.  Mens vi ser på en **meget lille MVC implementering** med **database connectivity**, gennemgår vi hvordan:  - Vi **seeder database**  - Hvordan vi skal **passe på med downloaded kode** fordi der ingen validering er og dermed sårbart overfor **CrossSite scripting (XSS)**  - Hvordan vi kan orientere os i en **webbrowser** på en webside **med developer tools (F12)**  - Setup til at bruge **XDebug Listen**, så du **i VS Code, kan trace hvad en bruger gør i en webbrowser.** Vi skal bruge flere debug processer som begge skal køre samtidigt.  **Noter**  W3school PHP Forms  <https://www.w3schools.com/php/php_forms.asp>  Cross-site Scripting (XSS)  <https://csrc.nist.gov/glossary/term/cross_site_scripting>  Settings i apache php.ini  [XDebug]  xdebug.remote\_enable = 1  xdebug.remote\_autostart = 1  xdebug.mode = debug  xdebug.start\_with\_request = yes  zend\_extension = C:\xampp\php\ext\php\_xdebug.dll |

|  |  |
| --- | --- |
| **Client server Table og CRUD**  (57 minutter)  **Website:**  **MySQL PHP MVC CRUD Without Framework**  <https://www.c-sharpcorner.com/article/mysql-php-mvc-crud-without-framework/>  **Screencast**  <https://www.studietube.dk/video/7478660/9325279331c4551a19a896416197b0c9>  **Git**  [**https://github.com/kimo1ucl/Database-Connectivity-Examples**](https://github.com/kimo1ucl/Database-Connectivity-Examples)  Folder name: Assistant mvclesson | **Indhold:**  Vi ser på hvordan man i en MVC implementering trinvist kan inkrementere kode, for at tilføje funktionalitet. I dette eksempel bare en tabel, der i starten bare oplister, dernæst implementeres i iterationer CRUD. Vi anvender poweshell til gennemgangen, og bruger powershell til at seede database, til at deploye kode til vores ”webwerver”, og får startet en internet browser op vi kan bruge til at teste brugeractions, og en instans af VS Code, der peger ned i koden der kører i browseren.  **Noter**  00:00-03:25: Intro – Inkrementere kode, sprint for sprint  03-25-06:40: Kodeændringerne i relation til implementerings rækkefølgen  06:40-12:25: Et hurtigt kig ned i powershell og seed af database + tools vinduer.  12:25-14:45: Gennemgang af List kode  14:45-17:55: Debug af List kode  17:55-19:30: Opsætning og gennemgang af Insert Kode  19:30-25:30: Debug af Insert kode  25:30-28:30: Recap: List og Insert  28:30-29:30: Gennemgang af Delete ændringer  29:30-32:50: Insight powershell: ”Deploy til website” (filecopy) + Tools kommandoer  32:50-34:45: Afprøvning af Delete UI  34:45-37:45:Forberedelse debug af Delete kode  37:45-44:15: Debug af Delete kode (inkl. underlig VS kode fejl -> Prøv at genstarte VS Code  44:15-48:30: Forskellen på view/Insert.php og view/update.php, og hvorfor egentlig i separate filer.  48:30-56:44: Debug af Update kode |