0.4 - Advanced Concepts

MySQL 101

Episode 11 is about how data is stored in a (MySQL) database and how you can use the terminal to access this data. Besides the terminal, Jeffrey shows other applications that allows you to work with the same data, like TablePlus, NaviCat and SequelPro. Before you watch the episode, we first define some terms that you need to know first.

Basics

The next video explains some basic database concepts:

Conclusion: MySQL is a relational DBMS and SQL is used to query a database.

> 💡 The "M" in XAMPP stands for "MySQL". So, when you installed XAMPP, MySQL is installed with it. You can start the application from the XAMPP control panel. However, when you do this, nothing seems to happen. This is because most DBMS applications operate in the background. You can 'connect' to the MySQL instance using IP address 127.0.0.1 and port number 3306, but you need another application to do this.

Tables

Relational databases are organized in tables. Each database consists of one or more tables. This is basically like an Excel spreadsheet, only with one difference: each table has its own specific set of columns and each column holds a specific attribute of something you want to store information about.

A database about music might consist of a table that holds information about artists. This table might have columns for attributes like the artists name, date of birth and country of birth. Each row in this table contains the values of each attribute of some artist. Other tables that might exist in this database are tables that holds song information or information about albums, etc.

MySQL and Databases

A typical MySQL instance can manage multiple databases. Each databases is completely separated from each other. Users that want to query a database must be granted privileges to that specific database. Only a database administrator user has access to all databases.

Database applications

Because MySQL runs in the background, you need another application to work with it. These applications might be:

- The terminal. With special commands, you can connect to MySQL, select a specific database and send SQL commands to query that database. Jeffrey does this in this episode
- Database Management applications like TablePlus, NaviCat or Sequel Pro. These applications allow you to do the same as the terminal, but in a more visual manner. XAMPP comes with a browser based application called PHPMyAdmin.

- Modules/plugins in programming platforms like PDO in PHP. These allow developers to build applications that uses a database to store its data. Jeffrey will show you how PDO works in episode 13.

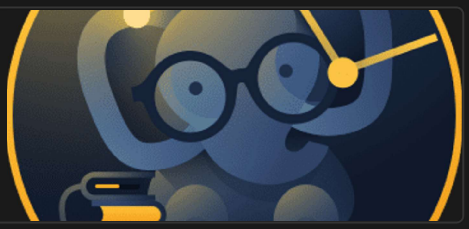
💡 Jeffrey uses the terminal to access the database. If you use XAMPP, this option is not available out of the box. If you want to do this, you must update the `PATH` variable again. But, because most developers use other applications for database management, you should only watch what he is doing in the video. As an alternative use the **Admin** button in the XAMPP config. This opens the PHPMyAdmin application.

Now, watch the episode.

MySQL 101

You've only just begun learning the fundamentals of PHP, and already I'm throwing something else at you. Sorry (bows head in

https://laracasts.com/series/php-for-beginners/episodes/11



👩‍💻 **Exercise 0.7** Start your MySQL server. Use (install if needed) a database management application of your choice to list all the databases. Jeffrey used the terminal application to do this (see the video time around 3:00). Compare your result with Jeffrey`s and note the differences.

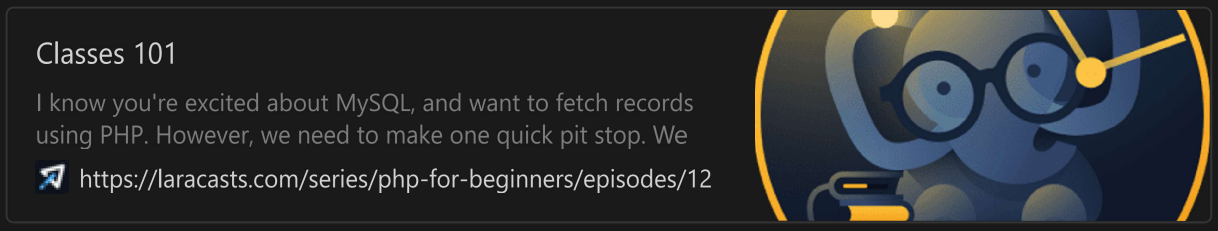
👩‍💻 **Exercise 0.8** Use your database management application to create the same database (mytodo) with the table (todo) with the same columns just as Jeffrey did, including the auto incrementing primary key. Almost every database management application supports this. You need to find out for yourself how you can perform this task with your chosen application. Insert two or more rows into the table and experiment with SQL to select one record.

Classes 101

PHP also includes Object Oriented concepts like classes, objects, attributes, constructors, methods and inheritance like you learned in the previous quarter. Jeffrey introduces this in the next video.

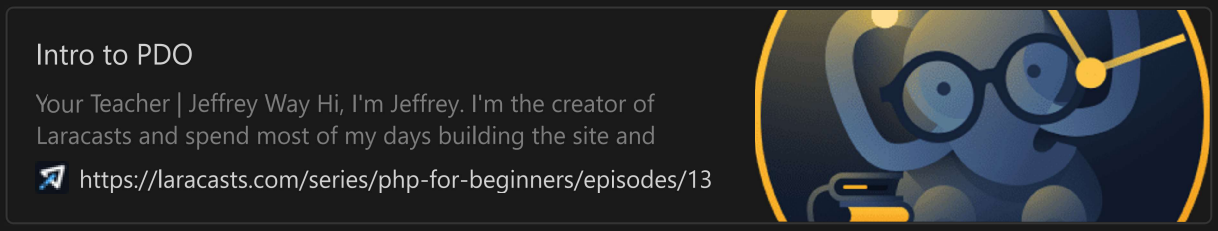
> 💡 Laravel uses a lot of OO principles like classes and inheritance. This helps to keep the code readable.

As you watch the video, you will notice that the OO basics are exactly the same, but there are some differences. The most noticeable will be that PHP uses the `->` syntax to call a method instead of `.`.

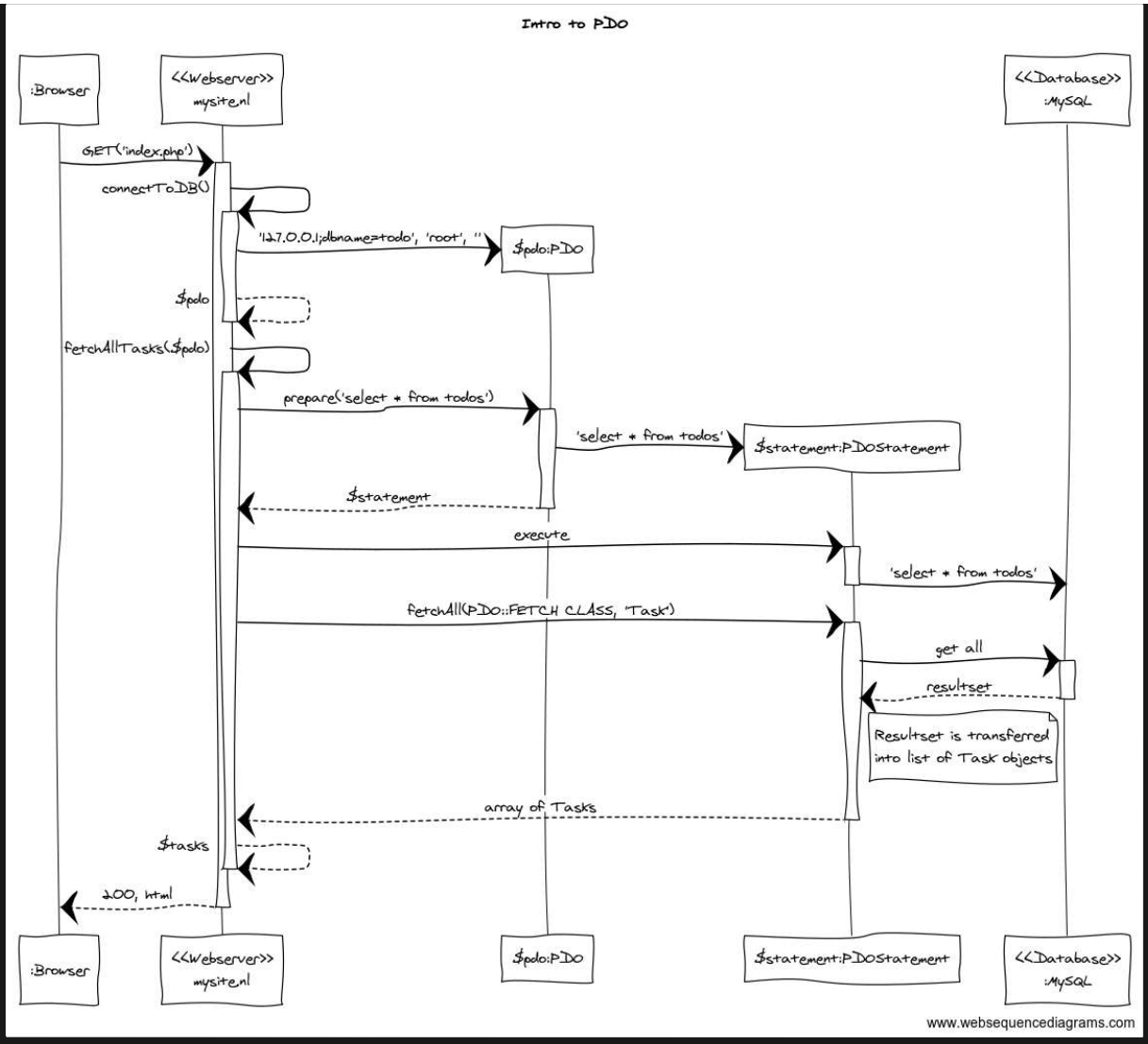


Intro to PDO

The next episode uses classes that already exist in the PHP platform to connect to a database, query a table and render the results in an HTML page.



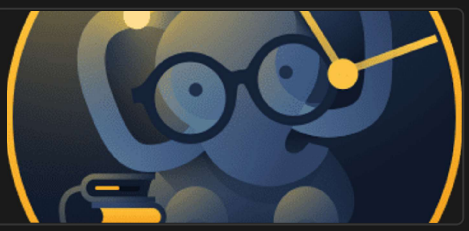
The sequence diagram below show a model of the final version of the code. Notice that you can see a MySQL instance as an object and the `$statement` object is an instance of the `PDOStatement` class.



Refactoring

More OO

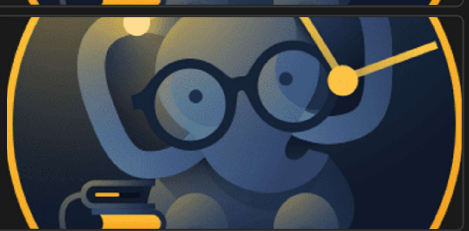
The next episodes refactors the existing code into a better readable structure using OOP concepts.



PDO Refactoring and Collaborators

In this episode, you'll perform your first refactor. "Refactor" is a term we use to describe the process of changing the structure of

https://laracasts.com/series/php-for-beginners/episodes/14



Hide Your Secret Passwords

At the moment, we're hardcoding our database username and password directly within the Connection class. But, we don't

https://laracasts.com/series/php-for-beginners/episodes/15

Alternative Routing

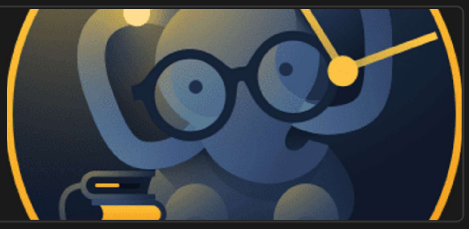
Up to now, we create a separate file for each HTTP request. The next video shows an alternative way to route HTTP requests to different PHP files.

> 💡 Routing is one of Laravel's core concepts. It is good to have a basic understanding of the basic mechanics of this.

Make a Router

Let's discuss routing in this episode. While, yes, you can create PHP files that correspond to the URI, this breaks down pretty

https://laracasts.com/series/php-for-beginners/episodes/16



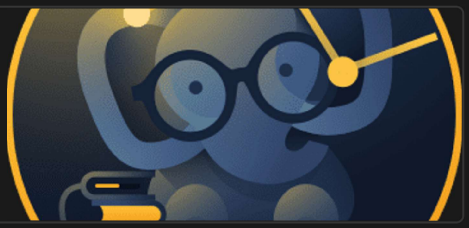
Cleaning up the views

Again, Jeffrey is refactoring his code. In this episode, he removes code duplications in the views.

Dry Up Your Views

Let's take some time to work on our views. At the moment, we've been rewriting the entire wrapping HTML for every single

https://laracasts.com/series/php-for-beginners/episodes/17



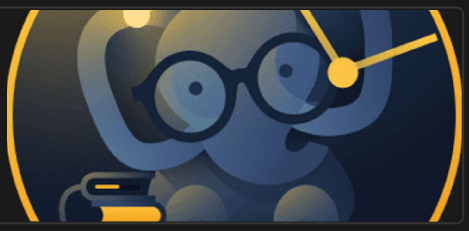
Array Filtering

The next episode reviews some array functions.

Array Filtering

PHP's various array functions are tools that absolutely need to rest in your belt. While the good ole'foreach can usually get the

https://laracasts.com/series/php-for-beginners/episodes/18



Forms, Request Types and Routing

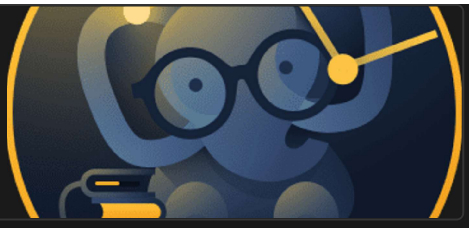
The next two episodes introduces the basics of how a user can send data to the server with information to add a new record in a table, let the server insert a new record and the redirect the user to another page.

> 💡 Jeffrey also introduces HTML forms, HTTP request types and Redirects. Just try to get a basic understanding of these subjects here, we will discuss this in more detail later.

Forms, Request Types, and Routing

Let's transition over to working with forms. If we want a form's submission to POST to a particular endpoint, well, our router

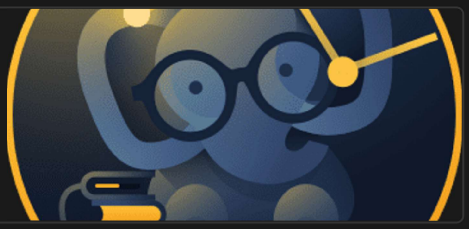
https://laracasts.com/series/php-for-beginners/episodes/19



Dynamic Inserts With PDO

So far, we've only written the logic to fetch records from the database. But now it seems that we need to, not only insert new

https://laracasts.com/series/php-for-beginners/episodes/20



Composer autoloading

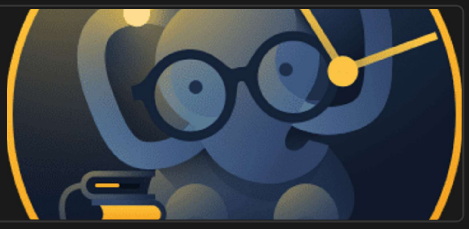
Jeffrey introduces Composer as the industry standard to handle packages and autoloading in PHP.

💡 Installing Composer is a must nowadays. Almost every PHP library or framework, including Laravel, needs Composer in some way.

Composer Autoloading

Your Teacher | Jeffrey Way Hi, I'm Jeffrey. I'm the creator of Laracasts and spend most of my days building the site and

https://laracasts.com/series/php-for-beginners/episodes/21



👩‍🔧 **Exercise 0.9** Use the terminal (try the `composer` command) to find out if composer is installed on your computer. If not, install and configure it until the `composer` command works like Jeffrey demonstrated (see the video around 1:27). See:

Composer

A Dependency Manager for PHP

https://getcomposer.org/

💡 For XAMPP user: The Windows installer of Composer will ask you the PHP path. Select your location of the `php.exe` file in your XAMPP installation.

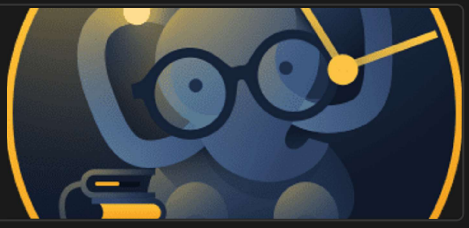
The Denouement, Or: How To Reinvent The Wheel

The last episodes refactors the code into a basic structure and concludes why it was not useful to repeat all of the code Jeffrey showed for you.

Your First DI Container

"Dependency injection Container" sounds like a super scary thing. But it's easy to understand! Think of them as boxes. Apply

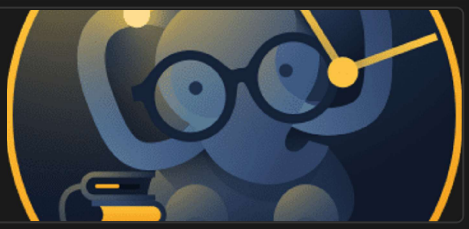
https://laracasts.com/series/php-for-beginners/episodes/22



Refactoring to Controller Classes

So far in this series, our "controllers" have simply been plain PHP files. In this episode, however, we'll refactor the router to allow

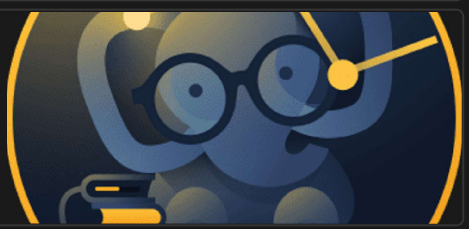
https://laracasts.com/series/php-for-beginners/episodes/23



Switch to Namespaces

At the moment, all of our classes share the same global namespace. But, think about it: you wouldn't throw your entire

https://laracasts.com/series/php-for-beginners/episodes/24



Meet Your Batteries Included Framework: Laravel

It's time for you to meet my framework of choice: Laravel (duh). In this lesson, we'll review the basic folder structure, discuss how

https://laracasts.com/series/php-for-beginners/episodes/25