Biblioteca Synanto v3

Desenvolvido por: Gabriel de Almeida

Tecnologias: PHP, JavaScript, HTML5, CSS3.

Padrão de projeto de software: MVC

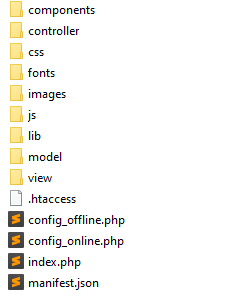
Data da primeira versão: 10/08/2020

Data da última versão: 12/01/2021

Synanto é uma Biblioteca que foi desenvolvida para agilizar o desenvolvimento de ferramentas, aplicações e sistemas Web baseados em PHP e JavaScript. Neste documento irei documentar e explicar os funcionamentos da biblioteca.

**Estrutura Básica**

A estrutura básica do Synanto é composta pelos componentes mostrados na imagem abaixo:



**Configuração**

Nos arquivos “config\_offline.php” e “config\_online.php”, contém as variáveis de ambiente (const). São utilizadas durante toda a aplicação.

SSL (bool)

PROTOCOL = SSL ? "https://" : "http://";

FOLDER (string)

URL = PROTOCOL."localhost/".FOLDER;

SERVIDOR = "mysql:host=localhost;dbname=NOME\_DO\_Banco;charset=utf8";

USUARIO = "root";

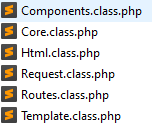
SENHA = "";

BASE\_TITLE = "sistema";

* SSL: Ativa ou Desativa o protocolo de segurança
* PROTOCOL: Recebe um string de acordo com o SSL
* FOLDER: Especifica em qual pasta o sistema está
* URL: Monta o caminho padrão a partir das outras variáveis de ambiente
* SERVIDOR: Configuração do servidor de banco de dados (Endereço do Servidor e Nome do Banco)
* USUARIO: Usuário do banco de dados
* SENHA: Senha do Banco de dados
* BASE\_TITLE: Usado para gerar o Nome da página assim: <title> BASE\_TITLE – Nome da View</title>

**Como ele funciona?**

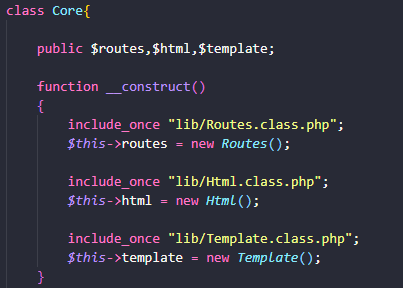
As classes contidas no diretório “lib/” são as que copila, entende e renderiza as páginas.



**Classe Core**

A classe “Core” é a principal. Ela realiza os includes das outras classes:

lib/Core.class.php



Com isso é possível realizar o “arranque” do Synanto:

lib/Core.class.php



Aqui obtemos da Classe “Routes” a rota principal.



Neste exemplo temos:

rota1 = Home (Rota Principal)

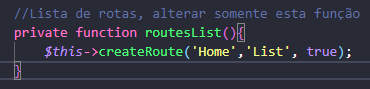
rota2 = Posts (SubRota)

rota3 = 8 (SubRota)

Neste caso, a view “Home.php” será renderizada apenas se ela estiver cadastrada na lista de rotas na classe “Routes”

**Classe de Rotas**

lib/Routes.class.php



Nesta classe é definido todas as rotas do sistema além de definir o “template” de cada rota e sua proteção.

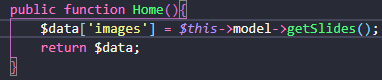
createRoute(ROTA, TEMPLATE, PROTEÇÃO)

- Template é o **conjunto** de Controller e Model, no exemplo da imagem acima, a rota “Home” utiliza List como template, logo dentro da pasta Model temos uma classe “ListModel.class.php” e dentro da pasta Controller temos uma classe “ListCtl.class.php”, e dentro da ListCtl temos um **função** Home() e obrigatoriamente é o nome da rota.

(O nome List é opcional ficando a critério do desenvolvedor criar quantos templates ele preferir)

**Controller**

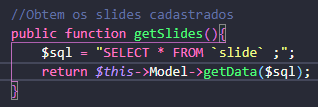
controller/ListCtl.class.php



Realiza a ligação entre View e Model na controller definimos as variáveis que irão para a View No exemplo acima na View Home.php a variável $images possui o resultado da função getSlides() que está na model, logo tudo dentro do array $data se transformará em variável quando a View é renderizada.

**Model**

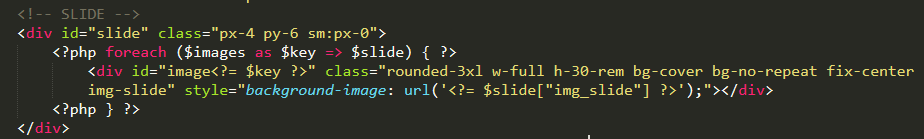
model/ListModel.class.php



Realiza a conexão com o banco de dados, aqui realizamos consultas que podemos exibir na View.

**View**

View/Home.php



Exibe o conteúdo HTML para o usuário.

**Classe Html**

lib/Html.class.php



Classe realiza a renderização da página realizando os imports da Head,Menu,View e Footer, Gerencia as dependências js e css.

**Classe Components**

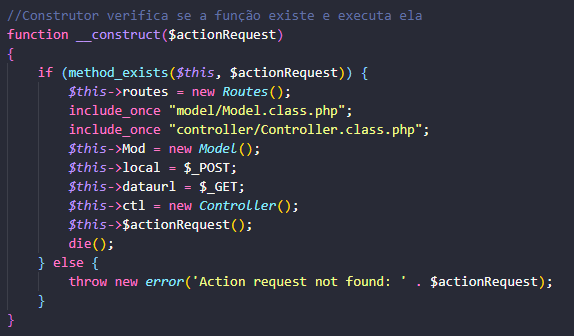
lib/Components.class.php



Utilizado para importar um componente a qualquer View (Ex: Menu, Footer) Normalmente ele fica no diretório componentes/

**Classe Request**

lib/Request.class.php



Realiza todas as ações do sistema através de requisições.