

Objetivo

Explicar o funcionamento, estrutura e objetivo do recurso de autorefresh na aplicação Multi24h no PHP

Objetivo do recurso

O autorefresh é um recurso disparado do lado do cliente para o servidor da aplicação PHP em busca de informações importantes a serem alertadas ou atualizadas em tela para o usuário para o escopo geral da aplicação.

Funcionamento

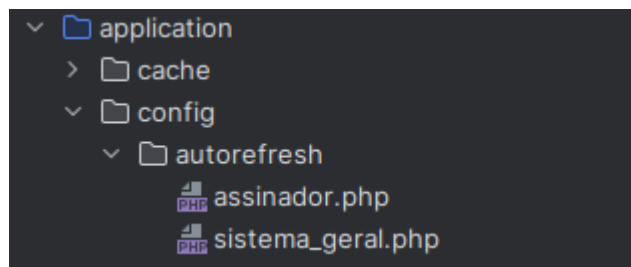
O frontend da aplicação através de um intervalo de tempo fica requisitando para o controller autorefresh e para o método task em busca de informações ou comandos para interação com a interface do usuário.

O sistema está configurado para efetuar essa requisição a cada minuto para o servidor, e cada módulo ou recurso pode definir internamente, de quanto em quanto tempo ele será executado e retornado para a interface através do método task.

Estrutura

Pastas para o backend

A estrutura de pastas e arquivos está disposta na pasta application/config/autorefresh, conforme imagem abaixo:



Onde fica cada arquivo de cada módulo do sistema que necessita de rotinas a serem executadas e retornadas com informação para a interface (frontend). O nome do arquivo segue o padrão já definido no sistema, conforme nome dos controllers dos módulos

Arquivos de módulo para o backend

```
$config['tasks'][] = [  
    'name' => 'sy_mensagens_usuario',  
    'interval' => 2,  
    'dispatch' => function () {  
        $CI =& get_instance();  
        $CI->load->library('message/UserMessage');  
        $CI->load->library('Session');  
  
        $user = $CI->session->userdata('userlogado');  
        $count = $CI->usermessage->unreadCount($user['codigo']);  
  
        return ($count > 0) ? $count : '';  
    }  
];
```

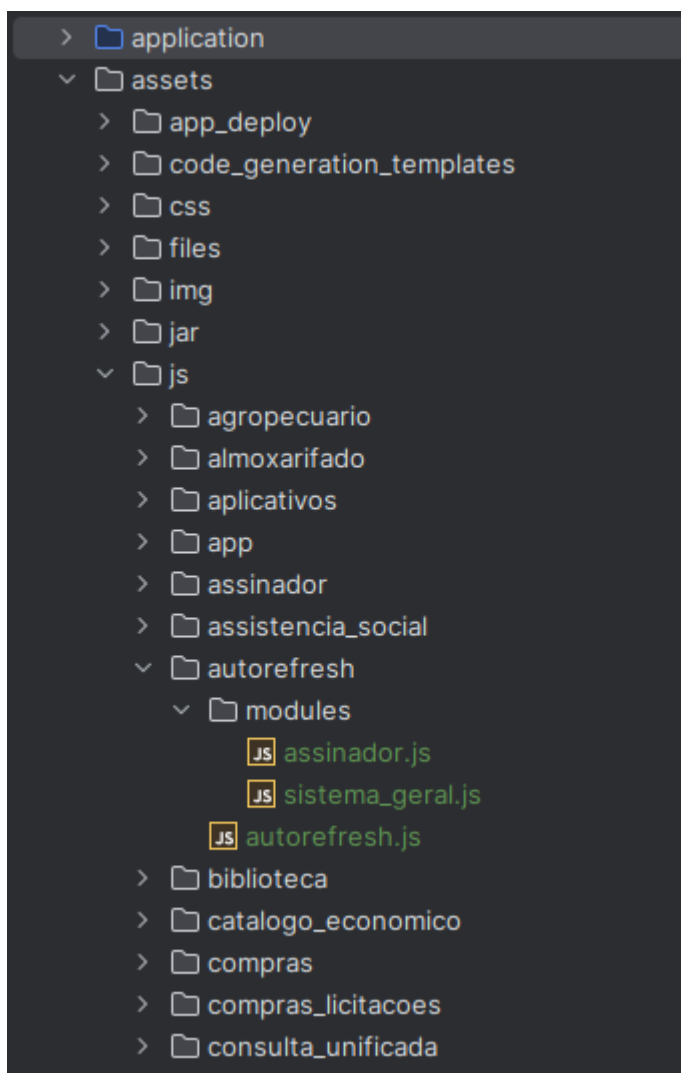
application/config/autorefresh/modules/sistema_geral.php

Acima está um exemplo de configuração de task para o autorefresh. A função dele é verificar a situação das rotinas em background e passar para a interface quantos jobs estão em execução e se há a necessidade de avisar o usuário a finalização de algum processo, sendo que:

- Todos os recursos devem ser alocados a variável \$config e na propriedade “tasks”, no formato de array multidimensional. Acima pode ser visto que foi definido dentro de \$config['tasks'][];
- Cada recurso, possui a estrutura de array, onde as propriedades:
 - name => Nome do recurso. Essa informação é obrigatória, ela será utilizada para ligar a rotina do servidor com a da interface;
 - interval => o intervalo de minutos que a rotina deve ser executada e retornada para a interface. Se não definido o padrão é execução instantânea a cada chamada do método task do controller autorefresh;
 - dispatch => Informação obrigatória, ela é a responsável por fazer a operação e retornar os dados necessários que serão utilizados na interface;

Pastas para o frontend

Para o consumo dos dados retornados pelo autorefresh vindos do servidor, é necessário registrar o evento no frontend. A organização dos arquivos se dá de forma similar ao registro do job no backend, conforme pode ser visto na estrutura de pastas abaixo:



Os arquivos JS dentro da pasta `application/assets/js/autorefresh/modules/` são carregados automaticamente na inicialização do sistema, assim registrando os eventos do autorefresh no frontend da aplicação.

Arquivos de módulo para o frontend

Por convenção e organização, a estrutura de arquivos segue o mesmo padrão dos outros recursos do sistema, ou seja, um arquivo para cada módulo do sistema Multi24h.

Cada arquivo de módulo será responsável por registrar/adicionar na classe `AutoRefresh` o controle dos eventos recebidos pelo backend.

Abaixo exemplo de um evento de autorefresh registrado no arquivo do módulo sistema geral, responsável por atualizar em tela o indicador de mensagens de usuário recebidas.

```
AutoRefresh.add('sy_mensagens_usuario', function (data) {  
    $('#menu-system-messages .badge.msg-counter').html(data);  
});
```

assets/js/autorefresh/modules/sistema_geral.js

Onde:

- AutoRefresh => Classe responsável por gerenciar os eventos de autorefresh recebidos do servidor de aplicação Multi24h no PHP
- add => Método responsável por registrar o evento, onde:
 - Primeiro parâmetro => nome do evento recebido do servidor, deve ser o mesmo nome registrado na propriedade "name" no evento da \$config
 - Segundo parâmetro => Função a ser executada quando o evento for recebido do servidor pelo autorefresh. O parâmetro "data" da função trata-se do dado recebido do servidor, que pode ser de qualquer tipo, seja, string, integer, number, object, etc.

Considerações finais

- O autorefresh está configurado para efetuar uma comunicação com o servidor de minuto a minuto em busca de dados dos eventos registrados, mas cada módulo, através do registro de eventos em [application/config/autorefresh/modules](#) pode definir individualmente em cada evento qual a periodicidade em minutos que ele será executado através da propriedade "interval" já mencionada neste documento;
- O autorefresh **NÃO DEVE** ser utilizado para apenas executar rotinas automáticas no servidor, para isso já existe o crontasks em [application/config/crontasks](#). Ele deve ser utilizado para ações onde é preciso de tempos em tempos ler informações do servidor e retornar para a interface efetuando alguma ação visual;
- Reforçando que o "name" do evento do lado do servidor precisa ser igual ao registrado em AutoRefresh.add(), para que ele possa ser lido e executado no lado do frontend;