

Бытует мнение и оно в принципе верное что использовать SSI и PHP в одном шаблонизаторе, есть нонсенс. Однако, существовало (и таки реализовано) ТЗ, в котором одним из требований было совмещение паттернов и инструкций SSI. Оставлю в стороне “законность” такого требования/решения и опишу что в результате получилось. Далее подкатом ...

Под паттерном (называйте как хотите) в данном контексте подразумевается выражение вида `<%menu%>` или `#menu#`, которое присутствует в темплейте страницы и позволяет в некотором роде отделить логику от HTML кода. То есть, проще говоря имеем обработчик, темплейт и паттерны. В зависимости от модуля (скажем страница категории), обработчик берет нужный темплейт, выдирает из него паттерны и заменяет их собранными кусками кода, которые затем и выводятся в виде полноценной страницы.

Так вот в эту схему слабо вписываются инструкции SSI. Имеем подводный камень (камни?) - расширение страницы, чтобы сервер правильно обработал SSI, должно быть **.shtml**

, а значит речь о PHP идти не может в принципе. Таким образом, если в нашем салате есть шаблонизатор с обработкой паттернов и SSI документы, которые нужно включить в страницу - представляется совсем грустная картина. Либо мы должны сначала собрать SSI страницу и затем заместить паттерны кусками кода, либо наоборот собрать темплейт из паттернов, сохранить как постоянный документ на сервере с нужным расширением и затем вызвать для подключения SSI инклюдов.

Готовое решение заключается в следующей последовательности действий:

- собственно **фишка**: в темплейте создаем паттерн вида:

```
<%include_virtual('/cgi-bin/script.cgi?ssi=1&'._SERVER['QUERY_STRING']);%>
```

- берем нужный темплейт с расширением **.php** , который содержит паттерны и может содержать PHP инклюды (`include_once('header.php')`) и не выводя в браузер берем из буфера:

```
ob_start();
include(TEMPLATES.$module['name'].'.mod.php');
$page=ob_get_contents();
ob_end_clean();
```

Делаем собственно с этим кодом, все что нам заблагорассудится, как то - меняем паттерны и пр.

- для вышеупомянутого паттерна, содержащего последовательность `include_virtual` используем правило замены с использованием замечательной функции [virtual\(\)](#) , в данном конкретном случае паттерн очищался от `<%include_%>` и оставшаяся часть исполнялась с помощью `eval`:

```
eval("virtual('/cgi-bin/script.cgi?ssi=1&'. $_SERVER['QUERY_STRING']);");
```

Получился совмещенный с SSI инклюдами шаблонизатор.