

# Kodmyran Front Overview

Author Kodmyran AB

## Introduction

Kodmyran Front is the frontend of Kodmyran Commerce and presents the actual HTML code to endusers. This section presents you with a short background of the typical operation of a Kodmyran Front template.

Each template is a collection of resources and code. All code is written either in PHP (server-side) or Javascript (client-side).

## Files & Directories

Each template in Kodmyran Front is stored in the `/templates` directory, each subdirectory contains a template; the name of the subdirectory is also the name of the template.

Avoid using template names with spaces, ampersands and other special characters. It will function but it complicates paths etc inside the template later.

Inside the template there can be a number of standard sub-directories

---

Directory	Purpose
snippets	This is the main directory for PHP code, all server-side code must be stored in this directory.
scripts	This is the main directory for Javascript code. It is used for client-side code.
css	The various stylesheets
mails	Template files used for sending e-mails
pages	An older directory used for page templates. Obsolete.
images	Used to store images used by the template, should not contain e.g. product images

---

At the top-level of the template directory there must be a file named `template.xml`, this file is the magic glue that informs the platform about the capabilities of the template. It indicates which stylesheets to reference, what scripts to use, the

API version the template supports and the supported languages.

All code within the template directory must use ISO-8859-1 encoding, you can not use UTF-8 at this time

## Recommendations

To create your own template we recommend using one of the systems standard templates and then copy the entire directory to a new name. Select the new name carefully and never start the template name with `kodmyran` as this name space is reserved for Kodmyran.

During an upgrade all `kodmyran\*` templates will also be automatically overwritten by their updated versions.

## Snippets

The term *snippet* refers to a small piece of code that serves a specific purpose, and it almost always refers to rendering some form of graphical HTML output. There is a special class/interface that Kodmyran Front uses to invoke snippets based upon information gathered from the `template.xml` file and the type of incoming page.

## Thread of execution

Upon receiving an incoming request Kodmyran Front will determine which template to use, load it, and then dispatch control to the `draw()` method of a snippet named `genericPage` (must be stored in `snippets/genericPage.class.php`). This class must extend the `shoppingSnippet` abstract class and implement at least the `draw()` method.

```
class genericPage extends shoppingSnippet
{
    public function draw( $shop, $name, $args )
    {
    }
}
```

The argument `shop` refers to an `iShop` object that is the top-most class container for Kodmyran Front. `Name` is the name of this instance of this class, it is most commonly “`genericPage0`” and is generally not used for this particular object.

The args variable consists of a string that indicates the type of page to draw, category, product, index etc. Most commonly there will be logic inside generic-Page.class.php to include another piece of PHP code for drawing that particular type of object.

You can also use the template object to instantiate and draw objects.

```
// Fetches the current template object
$template = $shop->getTemplate();

// Loads a new snippet and instantiate it (snippets/myPage.class.php)
// The new instance is named myPage0
$template->loadSnippet( "myPage0", "myPage" );

// Draw the snippet, and send the string "arg" as an argument
$template->drawSnippet( "myPage0", "arg" );
```