

آموزش فریم ورک گتنا نسخه 1.7.GFB

فریم ورک گتنا سازگار با PHP 5.5



گروه تخصصی نت افزار

یافته‌های را با باخته‌های مقایسه کن اگر "خدا" را یافتی هر چه باختی مهم نیست. گوش‌های "خدا" پر است از آرزو و دست‌هایش پر از معجزه؛ شاید بزرگترین آرزوی تو کوچکترین معجزه "خدا" باشد.

چه تغییراتی در 1.7.GFB حاصل شده است؟

۱۳۹۲/۰۴/۲۴

ویرایش چهارم

<http://www.gtna.net>

پاییز ۱۳۹۱

توجه: جهت راحتی طراحان فیلم‌های متعددی در رابطه با یادگیری این فریم ورک بر روی سایت منتشر شده است و کاملاً رایگان است. لطفاً از محصولات ایرانی حمایت کنید ما را با نام **طراحی سایت با فریم ورک گتنا** لینک کنید. نویسندگان: سید هادی منصوری راد – عصمت کارگر



کاربرد فریم ورک گتتا

نحوه پیکربندی فایل GTNA_Config

روش جلوگیری از ذخیره سازی صفحات

روش جلوگیری از ارسال فرم های خارجی

روش ایجاد یک Controller

روش ایجاد یک View

روش ایجاد یک Model

چگونه از URL ها استفاده کنیم؟

چگونه از Library ها استفاده کنیم؟

شیوه استفاده از کتابخانه GTNA_URL

چگونه با استفاده از کتابخانه GTNA_Captcha سؤال امنیتی ایجاد کنیم؟

پیاده سازی کد کپچا با استفاده از Session

پیاده سازی کد کپچا با استفاده از جدول کپچا

چگونه از کتابخانه GTNA_Session استفاده کنیم؟

GTNA_Session->open()

GTNA_Session->register()

GTNA_Session->read()

GTNA_Session->unregister()

GTNA_Session->encode()

GTNA_Session->decode()

GTNA_Session->id()

GTNA_Session->new_id()

GTNA_Session->RBI()



GTNA_Session->SRARO()

GTNA_Session::sess_read()

GTNA_Session->sess_close()

چگونه از کتابخانه **GTNA_Security** استفاده کنیم؟

GTNA_Security->sanitize_string()

GTNA_Security->escape()

GTNA_Security->strip_tag()

GTNA_Security->encrypt()

Mysql_sanitize_string()

چگونه با متدهای بانک اطلاعاتی عملیات **CRUD** انجام دهیم؟

Mysql_open()

Mysql_close()

Mysql_insert()

Mysql_update()

Mysql_select()

Mysql_delete()

چگونه فرم‌های درون یک **view** را به یک متد خاص از کنترلر ارسال کنیم؟

روش استفاده از کتابخانه **GTNA_HTML**؟

اعتبار سنجی فرم‌ها به وسیله کتابخانه **GTNA_Validate**

روش طراحی آپلودر با کتابخانه **GTNA_Upload**

اعتبار سنجی حرفه ای فرم‌ها

افزایش سرعت فریم ورک گتتا

سیستم کشینگ به وسیله **GTNA_Cache**

ایجاد سبد خرید به وسیله کتابخانه **GTNA_Ecard**



فریم ورک گتتا نسخه 1.7.GFB

روش طراحی ماژول برای فریم ورک گتتا

نصب **Nusoap** در فریم ورک گتتا

نصب **Smarty** در فریم ورک گتتا



کاربرد فریم ورک گتتا

این فریم ورک برای افرادی طراحی شده است که قصد دارند خیلی سریع و آسان پروژه‌های خود را در حد متوسط پیش ببرند. البته گتتا (گروه تخصصی نت افزار) قصد دارد در آینده‌ی نه چندان دور، قابلیت‌های گسترده‌ای را به این فریم ورک اضافه کند. اکنون در حال بررسی و تحقیق بر روی هوش مصنوعی و ترکیب آن با این فریم ورک می‌باشیم. اگر بخواهیم به توانایی‌های آن اشاره کنیم، یکی از ساده‌ترین فریم ورک‌ها از لحاظ یادگیری است. شما تنها در یک روز می‌توانید با کلیه‌ی متدهای آن آشنا شوید و پروژه‌های خود را به وسیله‌ی آن طراحی کنید.

چرا باید از فریم ورک استفاده کنیم؟

دلایل زیادی وجود دارد که خیلی از طراحان سایت‌ها و اپلیکیشن‌های تحت وب را به سمت فریم ورک‌ها سوق داده است. ما به چند مورد اشاره می‌کنیم:

- سایت‌ها و اپلیکیشن‌های تحت وب در یک چهارچوب مشخص و با نظم خاصی طراحی می‌شود.
- می‌توانید از کلاس‌هایی که از قبل طراحی شده، استفاده کنید.
- طبق استانداردهایی که برای فریم ورک طراحی شده است، به فایل‌ها و پوشه‌ها نظم می‌دهد.
- به دلیل طراحی سایت بر پایه معماری سه لایه، امنیت پروژه‌هایتان را افزایش می‌دهد.
- تولید و توسعه پروژه‌هایتان را تا حد خیلی زیاد آسان می‌کند.
- URLها را تا حد ممکن کوتاه می‌کند.
- به راحتی می‌توانید از طریق URL به کنترلرها و متدهای آن دسترسی داشته باشید و همچنین متدها را با مقدار فراخوانی کنید.

البته این موارد محدودیتی در کدنویسی ایجاد نمی‌کند. شما به راحتی می‌توانید کلاس‌های خود را به عنوان یک کتابخانه به فریم ورک اضافه و از کلاس‌های موجود این فریم ورک استفاده کنید و متدهای آنها را فراخوانی نمایید.

آیا این فریم ورک پشتیبانی مداوم دارد؟

جواب به این سؤال بستگی به کاربران دارد. اگر گتتا به وسیله کاربران مورد حمایت قرار بگیرد صد در صد این پروژه تا زمانی که گتتا وجود دارد بروزرسانی می‌شود و در خدمت طراحان ابزارهای تحت اینترنت قرار



خواهد گرفت. گتتا تمام سعی و تلاش خود را برای بهبود عملکرد این فریم ورک می کند تا نیازهای طراحان را در حد وسیع مرتفع سازد.

نحوه پیکربندی فایل GTNA_Config.php:

شما به وسیلهی آدرس زیر به این فایل دسترسی دارید و می توانید تنظیمات اولیه را برای طراحی پروژه هایتان را انجام دهید:

application/config/GTNA-Config.php

system_controller="welcome": این متغیر برای مشخص نمودن کنترلر پیش فرض می باشد. یعنی اگر قصد داشته باشید هنگام بارگذاری سایت در مرحله اول، کنترلر دلخواهتان باز شود کافی است که به جای **welcome** نام کنترلر خود را وارد نمایید. در این فریم ورک به صورت پیش فرض، کنترلر **welcome** بارگذاری می شود و اگر نام کنترلری وارد نمایید که وجود نداشته باشد پیام زیر برای شما نمایش داده می شود.

Controller dose not exist.

مرحله بعدی برای تنظیم فریم ورک تنظیمات مربوط به پورت، آدرس سایت و شاخه ای است که شما پروژه خود را در آن قرار داده اید و می خواهید بر روی آن **URL** کار کنید. به عنوان مثال فرض کنید شما قصد دارید به صورت زیر بر روی پروژه خود کار کنید:

http://127.0.0.1:81/GTNAF

فریم ورک گتتا برای چنین تنظیماتی، چندین متغیر در نظر گرفته است. به متغیرهای زیر توجه کنید.

//System root settings

```
$System_url="http://127.0.0.1;  
$System_directory="GTNAF";  
$System_port=81;
```

فریم ورک گتتا از تنظیمات بالا به این نتیجه می رسد که سیستم شما به صورت **local** می باشد و فایل های اصلی شما در شاخه ای به نام **GTNAF** قرار دارد و همچنین تمام آدرس ها به جز آدرس اصلی سایت یعنی **system_url** بر روی پورت ۸۱ فعال است. تنظیمات پورت به این دلیل است که شاید شما بخواهید بر روی پورتی به غیر از پورت ۸۰ کار کنید. بنابراین درک فریم ورک به صورت یک آدرس یا **URL** به شکل زیر است.



<http://127.0.0.1:81/GTNAF>

پس از گذر از این مرحله شما باید برای پروژه‌ی خود یک شاخه‌بندی در نظر بگیرید. مثلاً فایل‌های جاوا اسکریپتی و یا فایل‌های مربوط به استایل سایت در شاخه‌های مخصوص خود قرار بگیرند. فریم ورک گتتا برای کنترل و همچنین توسعه راحت‌تر پروژه‌ها این تنظیمات را در نظر گرفته است. همچنین شما با این کار تنها با یک متغیر به فایل‌های خود در **view** دسترسی دارید. (کوتاه سازی آدرس‌های) متغیرهای زیر را در نظر بگیرید:

```
$css_url=SYSTEM_URL.$system_file_directory."styles";
$java_url=SYSTEM_URL.$system_file_directory."java";
$jquery_url=SYSTEM_URL.$system_file_directory."jquery";
$image_url=SYSTEM_URL.$system_file_directory."images";
$other_url=SYSTEM_URL.$system_file_directory."other";
```

styles-java-images-other نام‌شاخه‌هایی است که در مسیر **application** قرار دارد. یعنی درک سیستم برای فایل‌های استایل و ... به شکل **URL** به صورت زیر است:

برای فایل‌های استایل:

<http://127.0.0.1:81/GTNAF/application/styles>

برای فایل‌های جاوا:

<http://127.0.0.1:81/GTNAF/application/java>

برای فایل‌های تصویری :

<http://127.0.0.1:81/GTNAF/application/images>

برای فایل‌های نامشخص :

<http://127.0.0.1:81/GTNAF/application/other>

اگر به عنوان مثال، یکی از کدهای بالا را به شکل زیر تغییر دهید

```
$css_url=SYSTEM_URL.$system_file_directory."my-styles";
```

باید یک دایرکتوری در مسیر **application** با نام **my-styles** ایجاد کنید. بعد از انجام این تنظیمات از طریق ثابت‌های زیر، در **view**ها به این مسیرها دسترسی دارید. البته نام فایل‌ها را خودتان باید به URL اضافه کنید.



CSS_URL

JAVA_URL

JQUERY_URL

OTHER_URL

به عنوان مثال از این پس زمانی که بخواهید یک فایل استایل موجود در دایرکتوری `style` و یا `my-styles` در یک `view` بارگذاری کنید کافی است تگ لینک را به شکل زیر بنویسید:

```
<link href="<?php echo CSS_URL/style.css ?>" />
```

برای بارگذاری فایل‌های دیگر هم به همین شکل می‌توانید عمل کنید.

حال ما فرض را بر این می‌گذاریم که شما می‌خواهید فایل‌های خود را در شاخه‌ای به غیر از شاخه‌های موجود قرار دهید. کافی است به شکل زیر عمل کنید:

```
$upload_url = SYSTEM_URL.$system_file_directory."upload";
```

به جای `upload` هر نام دیگری می‌توانید قرار دهید و برای اینکه این مسیر در کل `view`ها شناخته شود کد زیر را به فایل `GTNA_Config.php` دقیقاً زیر ثابت‌ها اضافه کنید.

```
define(UPLOAD_URL," $upload_url" );
```

از این پس شما به وسیله متغیر `UPLOAD_URL` به فایل‌های موجود در این دایرکتوری دسترسی دارید. کافی است مسیرهای زیر را به شکل زیر در یک `view` به خروجی ببرید تا متوجه منظورمان شوید.

```
echo CSS_URL;
```

```
echo JAVA_URL;
```

```
echo JQUERY_URL;
```

```
echo OTHER_URL;
```

و با تعریف دایرکتوری `upload` که خودمان به فایل `GTNA_Config.php` اضافه کردیم، به صورت زیر به آدرس آن دسترسی داریم:

```
echo UPLOAD_URL
```

`$_CS['STATE']`: این متغیر دو مقدار `ON` و `OFF` می‌پذیرد همچنین نسبت به حروف کوچک و بزرگ حساس است سیستم به صورت پیش فرض این متغیر را برابر با `OFF` قرار داده است با `ON` شدن این متغیر هیچ کس نمی‌تواند محتوای سایت شما را ذخیره کند بنابراین در حد چشم‌گیری امنیت را افزایش



1.7. فریم ورک گتنا نسخه 1.7.GFB

می دهد با تنظیم این امکان حتی توسط زبان های vb.net, C#.net, php, asp.net و... نمی توان محتوای سایت را ذخیره کرد.

\$_GP['STATE']: این متغیر دو مقدار **ON** و **OFF** را می پذیرد و نسبت به حروف بزرگ و کوچک حساس است. زمانی که مقدارش **OFF** باشد تمام فرم های ارسالی خارجی را می پذیرد اگر قصد دارید فرم های خارجی را **block** کنید می توانید مقدارش را برابر با **ON** قرار دهید می دانید که **block** کردن همه فرم های خارجی امکان دارد مشکلاتی را به همراه داشته باشد به عنوان مثال اگر شما بخواهید از درگاه های بانک استفاده کنید بانک به اجبار مقادیری ارسال می کند که شما باید آنها را بپذیرید بنابراین برای رفع این مشکل متغیر دیگری با نام **\$_LIST['ACCEPT']** وجود دارد در این متغیر لیست تمام دامین هایی که قرار است فرم های ارسالی از طرف آنها پذیرش شود را به صورت یک آرایه وارد می کنید توجه داشته باشید که حتما آدرس دامین خود را هم به لیست اضافه کنید در غیر این صورت حتی ارسال های داخلی هم **Reject** می شود.

توجه : دامین ها را بدون پرتکل **http** وارد نماید .

```
/* if $_GP['STATE']=ON then
*     $_LIST['ACCEPT']=array('www.gtna.net',
*                             'www.other.com',
*                             ...)
*/
$_LIST['ACCEPT']=array('127.0.0.1','www.gtna.net','www.other.com');
$_GP['STATE']=OFF; // GET & POST Accept state
$_GP['STATE']=ON;
```

تذکر : اگر مقدار متغیر **\$_GP['STATE']** برابر با **ON** باشد ولی متغیر **\$_LIST['ACCEPT']** خالی باشد با خطای زیر رو برو می شوید .

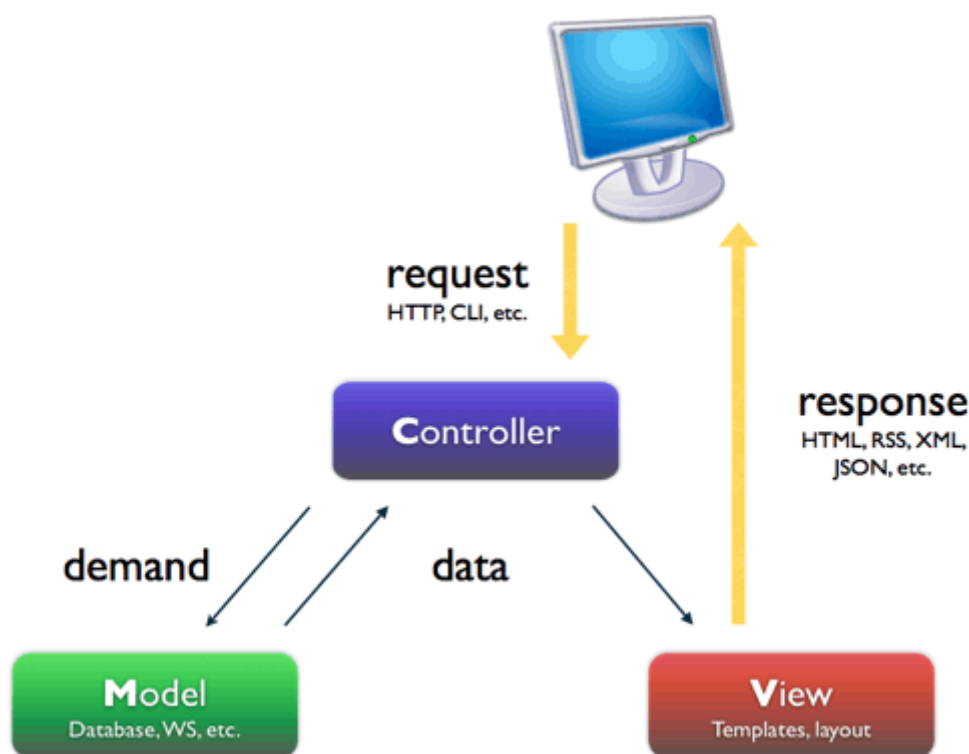
Please set your host url in **\$_LIST[ACCEPT]**
For EX:
====> **\$_LIST[ACCEPT]=array('www.gtna.net')**

گتنا قصد دارد در آینده نزدیک با طراحی یک الگوریتم رمز نگاری تمام داده های رد و بدل شده را رمز کند.



روش ایجاد Controller

اگر به شکل زیر توجه کنید، متوجه می‌شوید که معماری سه لایه یا three layer به شکل زیر عمل می‌کند.



مشخص است که درخواست از سمت کاربر به کنترلر ارسال می‌شود. کنترلر وظیفه دارد امنیت را برقرار کند. ما در این لایه می‌توانیم به لحاظ امنیتی داده‌ها را اعتبارسنجی کنیم، بنابراین کنترلر وظیفه پردازش داده‌ها و همچنین امنیت را بر عهده دارد. در صورتی که درخواست متقاضی توسط کنترلر مورد بررسی قرار گرفت، View مورد تقاضای کاربر توسط کنترلر نمایش داده می‌شود. اگر کاربر بخواهد داده‌هایی را از جداول درون پایگاه داده مورد جستجو قرار دهد باید این بار تقاضای خود را از طریق View به کنترلر ارسال کند. کنترلر بعد از انجام اعتبارسنجی و برقراری امنیت بر روی داده‌ها و یا اسکریپت‌سازی آن‌ها (معتبرسازی)، تقاضای کاربر را به لایه Model ارسال می‌کند. این لایه وظیفه دارد کلیه امور مربوط به بانک اطلاعاتی را انجام دهد. بنابراین لایه مدل کوئری‌های مربوطه را اجرا می‌کند و خروجی را به کنترلر ارسال می‌کند. توجه داشته باشید که این لایه تنها با لایه کنترلر مرتبط است سپس لایه کنترلر داده‌ها را به View ارسال می‌کند. بستگی به نوع سلیقه خودمان داده‌های مورد تقاضای کاربر را در هر جای صفحه وب که بخواهیم نمایش می‌دهیم. پس لایه کنترلر به عنوان یک واسطه عمل می‌کند و هیچ کاربری نمی‌تواند مستقیماً به یک View و



1.7.GFB فریم ورک گتتا نسخه

یا Model و حتی خود کنترلر دسترسی داشته باشد. یک فایل در سیستم وجود دارد که کتابخانه‌های مورد نیاز کنترلر را بار می‌کند و در صورت وجود، آنها را در کنترلرمان قابل دسترس می‌کند. از آن به بعد، کلیه داده‌ها توسط متدهای کتابخانه‌ها مورد بررسی قرار می‌گیرد و از لایه کنترلر به لایه‌های دیگر هدایت می‌شود. هر تقاضایی که از این دروازه عبور نکند به عنوان یک تقاضای نامعتبر در نظر گرفته می‌شود. بنابراین با کلیه این تفاسیر اگر شما بخواهید پروژه‌های خود را بر پایه معماری سه لایه اصلی یعنی MVC طراحی کنید، کار بسیار دشواری دارید گتتا این سکو را برای شما به ارمغان آورده است و شما به راحتی می‌توانید طبق معماری سه لایه پروژه‌های خود را مدیریت کنید. البته گتتا قصد دارد در آینده لایه هوش مصنوعی را به این لایه‌ها اضافه کند و این مورد باعث تمایز این فریم ورک نسبت به فریم ورک‌های دیگر می‌شود. البته به این مفهوم که این فریم ورک از فریم ورک‌های موجود بهتر است نمی‌باشد، ما تنها سعی کرده‌ایم تا تمام نیازهای اساسی مورد نیاز طراحان را به همراه امنیت به این فریم ورک اضافه کنیم و سادگی فراگیری آن را در نظر گرفته‌ایم. با حمایت شما انشالله در آینده قدرت فریم ورک را به رخ خواهیم کشید. همچنین قصد داریم سرعت پردازش اسکریپت‌ها را تا حد ممکن با ترفندهای خاص خودمان افزایش دهیم.

حالا به سراغ بحث اصلی خودمان می‌رویم برای اینکه یک کنترلر طراحی کنیم باید در گام اول کنترلرهای خود را در مسیر زیر ایجاد کنیم.

application/controllers/welcome.php

برای برقراری امنیت کنترلرها، کد زیر را به بالای کنترلر اضافه می‌کنیم.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
```

حالا با ایجاد یک کلاس دوست، اولین کنترلر را می‌نویسیم:

```
class welcome extends GTNA_controller{
function __construct(){parent::__construct();}
function index(){
    $this->load->view('welcome');
} // end index method
} // end welcome controller
```



نکته: زمانی که در مسیر `application/controllers` یک فایل با نام `welcome.php` و یا هر نام دیگری که ایجاد می‌کنید باید کلاسی که از `GTNA_controller` ارث‌بری می‌کند دقیقاً با همان نام باشد.

همانطور که مشاهده می‌کنید، ما یک کنترلر به نام `welcome` طراحی کرده ایم که از `GTNA_controller` ارث‌بری کرده است و شامل یک متد به نام `index` می‌باشد. توجه کنید که متد `index` یک متد الزامی است و در تمامی کنترلرها باید وجود داشته باشد.

توجه: به هر تعداد که تمایل دارید می‌توانید کنترلر ایجاد کنید، اما نمی‌توانید آنها را در دایرکتوری‌های متفاوت ایجاد کنید. در آینده این امکان به فریم ورک افزوده می‌شود.

درون متد `index` یک `view` بارگذاری شده است. این به این معنی است که در مسیر `application/views` فایلی به نام `welcome.php` قرار دارد. بنابراین شما باید از قبل این فایل را ایجاد کرده باشید در غیر این صورت با خطای زیر برخورد می‌کنید.

| application/views/welcome.php not found |

روش ایجاد View

کافیست یک فایل دیگر با نام `welcome.php` را در مسیر `application/views` ایجاد کنید و کدهای زیر را در آن قرار دهید. به این ترتیب `view` طراحی می‌شود.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="<?php echo CSS_URL."/style.css"; ?>" type="text/css" rel="stylesheet" />
<title>Untitled Document</title>
</head>
<style>
/* css code ... */
</style>
<body>
```



```
<?php
    echo SYSTEM_URL."<br>";
    echo CSS_URL."<br>";
    echo JAVA_URL."<br>";
    echo JQUERY_URL."<br>";
    echo OTHER_URL."<br>";
    echo IMAGE_URL."<br>";
    echo "welcome to GTNA framework."
?>
</body>
</html>
```

برای نمایش خروجی کافی است در مرورگر خود آی پی <http://127.0.0.1/GTNAF> را وارد نمایید، به همین ترتیب شما می‌توانید چند **view** را در یک کنترلر نمایش دهید.

اگر قصد دارید درون یک **view** چندین **view** دیگر بار کنید، از کد زیر استفاده نمایید.

```
load::view_is("view name");
```

روش ایجاد Model

قبلاً وظیفه‌ی لایه مدل را گفتیم. در اینجا قصد نداریم یک مدل ایجاد کنیم. جهت اتصال به دیتابیس و استخراج داده‌ها از درون جداول فقط به نحوه ایجاد یک مدل اشاره می‌کنیم و در آینده به تشریح متدهایی که در لایه مدل و همچنین تمامی متدهایی که در لایه کنترلر قرار دارد می‌پردازیم. شما باید تمامی مدل‌های خود را در دایرکتوری زیر ایجاد کنید:

application/models

اگر ما بخواهیم یک مدل با نام **gtna_model** ایجاد کنیم، باید ابتدا این فایل را به شکل زیر ایجاد کنیم.

application/models/gtna_model.php

```
<?php if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class gtna_model extends GTNA_model{
    function create(param){ your code ...} // method 1
    function read(param){your code ..} // method 2
    function update(param){your code...} // method 3
    function delete(param){your code...} // method 4
} // end gtna_model
```



اکنون شما مدلی به نام `gtna_model` که از `GTNA_model` ارث‌بری نموده است، ایجاد کرده‌اید که شامل ۴ متد `Create-Read-Update-Delete` یا `(CRUD)` می‌باشد اما کدنویسی‌های مربوط به آنها را در بخش کلاس‌های لایه مدل معرفی می‌کنیم و برای هر یک مثالی می‌آوریم تا با نحوه عملکرد و شیوه استفاده از آنها آشنا شوید.

چگونه از URLها استفاده کنیم ؟

URLها یکی از مهمترین قسمت‌هایی هستند که در فریم ورک‌ها کاربرد دارند. از طریق آنها می‌توان به کنترلرها و متدهای آنها دسترسی داشت و حتی می‌توانید متدهای کنترلرها را با مقدار فراخوانی کنید. برای مثال، فرض کنید که شما یک کنترلر به شکل زیر ایجاد کرده‌اید.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class show_message extends GTNA_controller{
    function display($message=NULL)
    {
        echo "This is 'GTNA framwork'";
    }
}
```

حال اگر بخواهیم از طریق `URL` به کنترلر و متدهای آن دسترسی داشته باشیم، شکل کلی آن به صورت زیر است.

<http://127.0.0.1/GTNAF/index.php/controller/method/value1/value2/.../value5>

بنابراین شما می‌توانید تا ۵ مقدار را به پارامترهای یک متد از کنترلر ارسال کنید اما در اینجا فقط یک پارامتر با نام `$message` داریم. انشالله در آینده قصد داریم کاری کنیم که شما بتوانید یک آرایه را از طریق `URL` دریافت کنید و همچنین چندین متد را از طریق `URL` همزمان فراخوانی کنید. حالا اگر بخواهیم به متد `display` از کنترلر `show_message` دسترسی داشته باشیم، باید به شکل زیر عمل کنیم (یعنی کافیسست که آدرس زیر را در مرورگر وارد نمایید).

http://127.0.0.1/GTNAF/show_message/display



اگر متد **display** دارای پارامتر باشد و ما بخواهیم آن را مقدار دهی کنیم به شکل زیر عمل می‌کنیم.

http://127.0.0.1/GTNAF/show_message/display/value1/.../value5

به جای value باید مقادیر مورد نیاز خود را ارسال کنید. در این صورت شما خروجی را بر روی مرورگر مشاهده خواهید کرد.

تذکر: در حال حاضر هیچ متدی نمی‌تواند بیشتر از ۵ پارامتر داشته باشد.

نکته: اگر کنترلری که از طریق URL فراخوانی می‌کنید وجود نداشته باشد، با پیام خطای زیر برخورد خواهید کرد.

Controller dose not exist

نکته : اگر همزمان با کنترلر متد آن را فراخوانی کنید و هیچکدام از آنها موجود نباشد با پیام زیر برخورد خواهید کرد (طبیعی است که وقتی کنترلری وجود ندارد، متدی از آن نیز وجود ندارد).

Controller and method dose not exist

توجه: حتی اگر آدرس صفحه اصلی یا home page خود را به شکل زیر فراخوانی کنید، با پیام خطا روبرو خواهید شد.

<http://127.0.0.1/GTNAF/index.php/>

در URL بالا یک اسلش اضافی وجود دارد. یعنی شکل درست آن به صورت زیر می‌باشد.

<http://127.0.0.1/GTNAF/index.php>

نکته بسیار مهم: شما باید در لایه کنترلر تمام مقادیری که از طریق URL به متدهای ارسال می‌شوند را به لحاظ امنیتی بررسی کنید. برای این کار می‌توانید با مراجعه با قسمت کتابخانه امنیتی GTNA_Security این کار را انجام دهید. در غیر این صورت هکرها با وارد کردن مقادیر نامعتبر از طریق URL اسکریپت‌های خود را به سایتتان تزریق می‌کنند. بنابراین تا زمانی که نتوانید با متدهای امنیتی فریم ورک کار کنید قادر به برقراری امنیت نمی‌باشید.



چگونه از Library ها استفاده کنیم؟

ما برای بارگذاری کتابخانه‌های موجود در این فریم ورک دو روش را در نظر گرفته‌ایم که به صورت زیر برای شما دسته بندی می‌کنیم.

۱- زمانی که قصد دارید از کتابخانه‌های اصلی موجود در هسته فریم ورک استفاده کنید، از کد زیر استفاده نمایید:

برای مثال :

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class Security extends GTNA_Controller
{
    function index()
    {
        $this->library('GTNA_Security');
        echo "<pre>";
        print_r( $this->GTNA_security->sanitize_string("help"));
    }
}
```

اگر به کنترلر **Security** توجه کنید، در متد **index** که به عنوان یک متد پیش فرض در نظر گرفته شده است، به صورت زیر کتابخانه امنیتی را فراخوانی کرده‌ایم.

```
$this->library('GTNA_Security');
```

و زمانی که بخواهیم از متد های **Security** و یا هر کتابخانه‌ی دیگری استفاده کنیم، بعد از دستور بالا به صورت زیر عمل می‌کنیم.

```
$this->GTNA_security->sanitize_string("help");
```

و چون خروجی آن از نوع آرایه می‌باشد، آن را با دستور **print_r** به خروجی برده‌ایم. قصد ما از مثال بالا تنها بارگذاری کتابخانه‌های موجود در هسته و نحوه استفاده از متدهای بود.

۲- زمانی که می‌خواهیم کتابخانه‌ای مورد نظر خودمان را بارگذاری کنیم و از متدهای آن استفاده نماییم.



1.7.GFB فریم ورک گتتا نسخه

اگر شما برای پروژه خود یک کتابخانه طراحی کرده‌اید و قصد دارید از متدهای آن در پروژه استفاده کنید، باید به صورت زیر عمل نمایید (فرض کنید کتابخانه‌ای به نام **Encode** ایجاد کرده‌اید که دارای یک متد به نام **string_encode** می باشد).

۱. در مرحله اول کتابخانه خود را در مسیر زیر قرار دهید.

Application/library/your library.php

Application/library/Encode.php

شکل کلی کدهای مربوط به کتابخانه شما باید به صورت زیر باشد.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class Encode {
    function string_encode($string=NULL){
        if(!$string)
            echo "please enter a string";
        else
            return md5($string);
        }
    }
```

۲. در مسیر زیر یک کنترلر با نام دلخواهتان ایجاد کنید و سپس به صورت زیر کد نویسی‌ها را انجام دهید.

Application/controllers/code.php

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class code extends GTNA_controller{
    function index()
    {
        $this->load->library(array('Encode'));
        $this->library('Encode');
        echo $this->Encode->string_encode ("GTNAF");
    }
}
```

اگر قصد دارید همزمان از چندین کتابخانه استفاده نمایید به صورت زیر اقدام نمایید



```
$this->load->library(array('Encode','Form',...));
```

شیوه استفاده از کتابخانه GTNA_URL:

اگر قصد دارید با استفاده از کتابخانه GTNA_URL یک دایرکتوری جدید برای فایل‌های خود ایجاد کنید ولی خارج از مسیر فریم ورک باشد. ابتدا باید در مسیر زیر یک دایرکتوری با نام دلخواهتان ایجاد کنید.

http://127.0.0.1/GTNAF/Your_directory_name

به عنوان مثال:

http://127.0.0.1/GTNAF/My_file

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class code extends GTNA_controller{
    function index(){
        $this->library('GTNA_URL');
        define('PATH_FILE', $this->GTNA_URL->system_url("My_file "));
    }
}
```

حالا در viewها می‌توانید از ثابت PATH_FILE استفاده نمایید. کافی است برای فهم بیشتر این موضوع بعد از ایجاد کنترلر بالا، کد زیر را در یک view تایپ کنید و خروجی آن را ببینید.

```
<?php echo PATH_FILE ; ?>
```

توجه: اگر دایرکتوری My_file توسط سیستم پیدا نشد با خطای زیر روبرو می‌شوید.

The requested URL /GTNAF/My_file/x.x was not found on this server

x . x نام و پسوند فایلی است که در مسیر My_file باید قرار داشته باشد، اما اکنون وجود ندارد.



چگونه با کتابخانه GTNA_Captcha سوال امنیتی ایجاد کنیم؟

قبل از اینکه به نحوه استفاده از کد کپچا بپردازیم کمی در مورد این کد و کاربرد آن توضیح می‌دهیم. در واقع این کد برای تشخیص انسان و ربات می‌باشد. ممکن است فرم‌هایی زیادی دیده باشید که در پایین آنها کدهایی وجود دارد که شما بعد از تکمیل نمودن آن فرم‌ها باید کد امنیتی را وارد نمایید در واقع با وارد کردن این کد، سیستم تشخیص می‌دهد که شما یک انسان می‌باشید یا یک ربات. در اینجا قرار نیست به نحوه سوء استفاده از فرم‌هایی که این کد را ندارد اشاره‌ای داشته باشیم. فقط این نکته را بدانید که کدهای امنیتی یا کد کپچا، امنیت را برای سایت هایتان به ارمغان می‌آورند. گتتا در حال حاضر تنها یک نمونه ساده، اما با امنیت بالادر یک کتابخانه به نام **GTNA_Captcha** برای شما فراهم نموده است.

در فریم ورک گتتا به دو روش می‌توانید کد کپچا ایجاد کنید

۱- کد کپچا به وسیله **Session** یا نشست

۲- کد کپچا به وسیله جدول **Captcha**

۳- کد کپچا ایجاد شده در نشست‌ها سرعت بسیار بالایی دارد اما امکان دارد هکر به وسیله ربودن نشست موفق به دسترسی به کد کپچا شود. برای جلوگیری از این امر در کنترلر نشست‌ها را رمزنگاری می‌کنیم ولی باز هم به طور کلی نسبت به روش دوم امنیت پایین‌تری دارد. طبیعتاً روش دوم امنیت بیشتری دارد، چرا که جواب سؤالات درون جدول کپچا ذخیره می‌شود و سپس با جوابی که کاربر ارسال نموده است مقایسه می‌شود. اگر پاسخ کاربر با جواب برابر باشد باید کد ذخیره شده درون جدول کپچا حذف شود این روش نسبت به روش نشست سرعت پایین‌تری دارد اما امنیت را تا حد بسیار بالایی افزایش می‌دهد.

۳. پیاده‌سازی کد کپچا با استفاده از **Session**

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class Login extends GTNA_controller{
function captcha()
{
    $this->library('GTNA_Session');
    $this->GTNA_Session->open();
    $this->GTNA_Session->encode();
    $this->library('GTNA_Captcha');
    $this->GTNA_Captcha->simple_captcha();
    $this->GTNA_Session->register('captcha',array('CR'=>CAPTCHA_RESULT));
    $this->load->view('login');
}
```



```
function form(){
    $this->library('GTNA_Session');
    $this->library('GTNA_URL');
    $this->GTNA_Session->open();
    if($this->GTNA_Session->RBI('captcha','CR')){
        if ($_POST['captcha_code']==$this->GTNA_Session->RBI('captcha','CR'))
            $this->load->view('admin');
        else
            $this->GTNA_URL->redirect("captcha");
    }else
        $this->GTNA_URL->redirect("captcha");
}
// end form method
// end of class
```

admin.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>welcome to administrator</title>
</head>
<body>
<center>welcome to admin</center>
</body>
</html>
```



login.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>login </title>
</head>
<body>
```

```
<form action="<?php echo SYSTEM_URL."index.php/Login/form" ?>" method="post">
<p><input type="text" name="user_name"></p>
<p><input type="password" name="user_pass" /></p>
<p><input type="text" name="captcha_code" /></p>
<p><?php echo CAPTCHA_CODE ?></p>
<p><input type="submit" value="login" /></p>
</form>
```

با اجرای پروژه، در صورتی که کد کپچا را درست وارد نمایید با پیام زیر روبرو می‌شوید.

Welcome to administrator

و در غیر این صورت، به صفحه login منتقل می‌شوید.

توجه: هیچ کاربری نباید بدون وارد نمودن کد کپچا وارد صفحه admin شود.

۴. پیاده سازی کد کپچا با استفاده از جدول کپچا

ابتدا جدول زیر را در دیتابیس خود import کنید.

```
--
-- Table structure for table `captcha`
--
CREATE TABLE `captcha` (
  `id` int(11) NOT NULL auto_increment,
  `answer` varchar(2) NOT NULL,
  `ip_address` varchar(15) NOT NULL,
  `expire` varchar(25) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=280 ;
--
-- Dumping data for table `captcha`
--
```



در مرحله اول کدکپچارا ایجاد می کنیم ونتیجه را درجدول کپچا ذخیره می نماییم. برای این کار باید کنترلر **comment** به صورت زیر ایجاد شود.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class comment extends GTNA_controller{
function index(){ $this->qv(); }
function library_load()
{
    $this->library('GTNA_Session');
    $this->library('GTNA_Captcha');
    $this->library('GTNA_Security');
    $this->load->model('captcha_model');
    $this->library('captcha_model');
}
function qv()
{
    $this->library_load();
    $this->captcha_model->delete_by_ip();
    $this->GTNA_Captcha->simple_captcha();
    $this->captcha_model->insert("20"); // تعیین زمان بر حسب ثانیه برای باطل شدن سوال
    $this->load->view('comment');
}
```

حالا نوبت به طراحی متد form برای کنترلر comment می رسد که سنگین ترین وظیفه را بر عهده دارد. در پایان توضیحات مختصری در مورد هر یک از متد ها می دهیم.



```
function form()
{
    $this->library_load();
    $captcha_answer=$this->captcha_model->search_by_ip();
    if(!$captcha_answer)
        $this->qv();
    else
    {
        if(time()>CAPTCHA_EXPIRE){
            $this->GTNA_Session->open();
            $this->GTNA_Session->register('message',"زمان پاسخ گویی شما به اتمام رسیده است");
            $this->captcha_model->delete_by_ip();
            $this->qv();
        }else
        {
            if($captcha_answer)
            {
                $captcha_code_user=$this->GTNA_Security->escape($_POST['captcha_code']);
                if($captcha_code_user==$captcha_answer)
                {
                    echo "پیام شما با موفقیت ثبت شد";
                    $this->captcha_model->delete_by_ip();
                }else
                {
                    $this->captcha_model->delete_by_ip();
                    $this->qv();
                }
            }else
            {
                $this->qv();
            }
        }
    }
}
```



```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class captcha_model extends GTNA_Model{
function insert($time=10)
{
    $data=array(
        'answer'=>CAPTCHA_RESULT,
        'ip_address'=>$_SERVER['REMOTE_ADDR'],
        'expire'=>time()+$time);
    $this->db->Mysql_open();
    $result=$this->db->Mysql_insert('captcha',$data);
    if($result)
        return $result;
    else
        return FALSE;
}

function search_by_ip(){
    $this->db->Mysql_open();
    $result=$this->db->Mysql_select('captcha',"ip_address='{$_SERVER['REMOTE_ADDR']}'");
    if($result){
        define('CAPTCHA_EXPIRE',$result[0]->expire);
        return $result[0]->answer;
    }else
        return FALSE;
}

function delete_by_ip(){
    $this->db->Mysql_open();
    $result=$this->db->Mysql_delete("captcha","ip_address='{$_SERVER['REMOTE_ADDR']}' limit 1");
    if($this->db->Mysql_delete("captcha","ip_address='{$_SERVER['REMOTE_ADDR']}' limit 1"))
        return TRUE;
    else
        return FALSE;
}
} // END MODEL
```



1.7. فریم ورک گتتا نسخه 1.7.GFB

دقت داشته باشید که نام کلاس مدل و نام فایل مدل باید دقیقاً مانند هم باشد این نکته را برای ایجاد کنترلرها نیز الزامی است.

بعد از طراحی مدل باید `view` را ایجاد کنیم برای ایجاد فرم `comment` ابتدا در مسیر زیر فایل با نام `comment` ایجاد کنید

Application/view/comment.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>comment</title>
</head>
<body>
<?php // کد پی ایچ پی زیر وظیفه دارد در صورت باطل شدن کپچا پیام ایجاد شده در نشست را به نمایش در آورد
    if(GTNA_Session::sess_read('message')):
        echo GTNA_Session::sess_read('message');
        GTNA_Session::sess_close();
    endif
?>
<form action="<?php echo SYSTEM_URL."index.php/comment/form"; ?>" method="post">
    <p><textarea cols="50" rows="10">this is a comment</textarea></p>
    <p><input type="text" name="captcha_code" /></p>
    <p><?php echo CAPTCHA_CODE?></p>
    <p><input type="submit" value="send" /></p>
</form>
</body>
</html>
```



مزیت‌های این روش :

- ۱- جواب کپچا در هیچ نشست ذخیره نمی‌شود. بنابراین هیچ کد کپچایی برای دزدی نشست وجود ندارد.
 - ۲- برای هر شخص، یک زمان پاسخ به سوال در نظر گرفته شده است. با گذشت زمان از حد مجاز حتی اگر پاسخ صحیح را وارد نماید کد کپچا به عنوان کد باطل شده تلقی می‌شود.
 - ۳- آی پی هر کاربر به عنوان یک رمز عبور برای پاسخ سؤالات در نظر گرفته می‌شود.
 - ۴- طوری طراحی شده است که هیچ افزونگی در جدول به وجود نیابد. یعنی برای هر شخص فقط در یک رکورد سوال ایجاد می‌شود. بنابراین هرچند کاربر به طور همزمان می‌تواند کامنت ارسال کند.
- این کد در عین سادگی به لحاظ امنیتی بسیار ایمن می‌باشد. ما فرض را بر این گذاشته‌ایم که یک ربات جواب سوال را پیدا کند اما باز هم در ارسال کامنت ناموفق خواهد بود.

چگونه از Sessionها (نشست‌ها) استفاده کنیم؟

جدول متدهای کتابخانه Session به صورت زیر است.

- ۱- `GTNA_Session->open()`: برای کار با نشست‌ها درون کنترلرها ابتدا کتابخانه‌ی نشست را به صورت زیر فعال می‌کنیم. حالا به تمامی متدها دسترسی دارید.

```
$this->library('GTNA_Session');
$this->GTNA_Session->open();
```

- ۲- `GTNA_Session->register()`: زمانی که بخواهید یک نشست با نام دلخواهتان ایجاد کنید به صورت زیر عمل کنید.

```
$this->GTNA_Session->register('your session name');
```

```
$this->GTNA_Session->register('CAPTCHA',$captcha_code);
```

- توجه:** به جای آرگومان دوم می‌توانیم هر نوع آرایه‌ای قرار دهیم (یک بعدی، دوبعدی و ...) و آن را درون نشست ذخیره کنیم.

- ۳- `GTNA_Session->read()`: برای دسترسی به نشست‌هایی که ایجاد کرده‌اید به صورت زیر عمل کنید.

```
Print_r( $this->GTNA_Session->read('CAPTCHA'));
```



توجه: مقدار برگشتی این متد از نوع آرایه می باشد.

۴- `GTNA_Session->unregister()`: برای باطل کردن نشست مورد نظرتان استفاده می شود. زمانی از این متد استفاده می کنیم که به عنوان مثال چند نشست را با نام های متفاوت ایجاد کرده ایم اما نمی خواهیم تمام نشست ها باطل شود. به این ترتیب به وسیله نام، نشست مورد نظر آن را از بین می بریم.

```
$this->GTNA_Session->register('TEST1','۱ آزمایش شماره');
$this->GTNA_Session->register('TEST2','۲ آزمایش شماره');
$this->GTNA_Session->register('TEST3','۳ آزمایش شماره');
$this->GTNA_Session->register('TEST4','۴ آزمایش شماره');
```

```
echo $this->GTNA_Session->unregister("TEST4");
```

۵- `GTNA_Session->encode()`: زمانی استفاده می شود که قصد داریم نشستمان را به صورت کد شده بر روی سرور ذخیره کنیم.

```
$this->GTNA_Session->encode();
```

توجه: این متد هیچ آرگومانی ندارد.

۶- `GTNA_Session->decode()`: اگر بخواهیم نشست ذخیره شده را از حالت کد شده به حالت اولیه بازگردانیم، باید به صورت زیر عمل کنید.

```
echo $this->GTNA_Session->decode('CAPTCHA');
```

به این ترتیب نشست کپچا از حالت کد به حالت اولیه باز می گردد.

۷- `GTNA_Session->id()`: برای دسترسی به کد Session مورد استفاده قرار می گیرد.

```
echo $this->GTNA_Session->id();
```

۸- `GTNA_Session->new_id()`: برای تغییر کد نشست مورد استفاده قرار می گیرد و نحوه استفاده از آن به صورت زیر است.

```
echo $this->GTNA_Session->new_id();
```



۹- `GTNA_Session->RBI()` مخفف **Read By Index** می باشد (زمانی استفاده می شود که می خواهیم به وسیله اندیس آرایه های ذخیره شده در نشست، به مقدار آن دسترسی داشته باشیم).

```
$this->GTNA_Session->register('captcha',array
    ('account'=>array('user_name'=>"gtna",'user_pass'=>"12345"),
    'profile'=>array('name'=>'hadi','fname'=>"mansoori rad"),
    'details'=>"30")
));
```

```
$data=$this->GTNA_Session->RBI('captcha',"details");
if($data)
    echo $data; // خروجی: ۳۰
else
    echo "not found";
```

```
$data=$this->GTNA_Session->RBI('captcha',"account","user_name");
if($data)
    echo $data;
else
    echo "not found";
```

خروجی: gtna

فرض کنید در اندیس `details` به جای 30، کلمه `world` را قرار داده ایم و می خواهیم به حرف `d` در کلمه `world` دسترسی پیدا کنیم.

W o r l d

0 1 2 3 4

```
$data=$this->GTNA_Session->RBI('captcha',"details",4);
if($data)
    echo $data;
else
    echo "not found";
```



۱۰- `SRARO:GTNA_Session->SRARO()` مخفف (session Read And Return Object) می

```
$this->GTNA_Session->register('captcha',array
    ('account'=>array('user_name'=>"gtna",'user_pass'=>"12345"),
    'profile'=>array('name'=>'hadi','fname'=>"mansoori rad"),
    'details'=>array('expire'=>"30")
));
```

اگر به مثال بالا توجه کنید، می بینید که یک آرایه سه بُعدی را درون نشست قرار داده ایم. برای اینکه بعدها را به صورت مجزا به نمایش در آورید، فقط کافی است نام آن بُعد را ذکر کنید. مثال زیر برای نمایش خروجی می باشد.

```
$data=$this->GTNA_Session->SRARO('captcha');
echo "<pre>";
print_r($data->account) ; // بعد اول
print_r($data->profile) ; // بعد دوم
print_r($data->details) ;// بعد سوم
print_r($data) ;// همه بعدها
```

اگر بخواهیم به یکی از فیلدهای آرایه درون آبجکت (\$data) اشاره کنیم، به صورت زیر عمل می کنیم.

```
echo "<pre>";
print_r($data->account['user_name']) ; // بعد اول شی دیتا آرایه اکانت فیلد نام کاربری
print_r($data->profile['name']) ; // بعد دوم شی دیتا آرایه پروفایل فیلد نام
print_r($data->details['expire']) ;// بعد سوم شی دیتا آرایه مشخصات فیلد اکسپایر
```

۱۱- `GTNA_Session->close()`: برای بستن کلیه نشست ها استفاده می شود

۱۲- `GTNA_Session::sess_read()`: زمانی که در یک کنترلر نشست ایجاد کرده اید و قصد دارید از آن نشست درون viewها استفاده کنید از این متد استفاده کنید.

توجه: این متد هم در view و هم در کنترلر قابل استفاده است.

۱۳- `GTNA_Session::sess_close()`: برای بستن نشست ها مورد استفاده قرار می گیرد.



جدول متدهای کتابخانه نشست به صورت زیر است:

GTNA_Session->new_id()	GTNA_Session->read()	GTNA_Session->open()
GTNA_Session->SRARO ()	GTNA_Session->decode()	GTNA_Session->close()
GTNA_Session->RBI()	GTNA_Session->encode()	GTNA_Session->register()
GTNA_Session::sess_read()	GTNA_Session->id()	GTNA_Session->unregister()
GTNA_Session::sess_close()		

چگونه از کتابخانه GTNA_Security استفاده کنیم؟

برای استفاده از متدهای کتابخانه ابتدا باید کتابخانه آن را بارگذاری کنید تا بتوانید از متدهای آن استفاده کنید.

۱- متد `sanitize_string()` دو پارامتر می‌پذیرد شکل کلی آن به صورت زیر است.

Sanitize_string(value,sanitize type);

اگر این متد را تنها با یک متغیر فراخوانی کنید و مقدار آن را ۱ یا **help** بگذارید، انواع فیلترگذاری بر روی داده‌ها را مشاهده می‌کنید. به قطعه کد زیر و خروجی آن دقت کنید.

```
$this->library('GTNA_Security');
Print_r($this->GTNA_Security->sanitize_string("help"));
```

خروجی:

```
Array
(
    [0] => [The sanitize type]
    [1] => FILTER_SANITIZE_MAGIC_QUOTES
    [2] => FILTER_SANITIZE_SPECIAL_CHARS
    [3] => FILTER_SANITIZE_NUMBER_INT
    [4] => FILTER_SANITIZE_EMAIL
    [5] => FILTER_SANITIZE_ENCODED
    [6] => FILTER_SANITIZE_NUMBER_FLOAT
    [7] => FILTER_SANITIZE_STRING
    [8] => FILTER_SANITIZE_STRIPPED
    [9] => FILTER_SANITIZE_URL
)
```



1.7. فریم ورک گتتا نسخه 1.7.GFB

اگر در مورد نحوه عملکرد فیلترهای بالا اطلاعاتی ندارید، باید دانش خود را افزایش دهید تا تشخیص دهید چه زمانی از این فیلترها استفاده نمایید.

یک مثال ساده برای نحوه استفاده از فیلترهای بالا:

```
$data="admin@gtna.net<>\\<script>";
echo $this->GTNA_Security->sanitize_string("$data","FILTER_SANITIZE_EMAIL");
```

متد **strip_tag()** به وسیله این متد فقط تگ‌هایی اجرا می‌شود که شما به آنها مجوز داده‌اید. به کد زیر توجه کنید.

```
$tag="<p>this is a test</p><strong>GTNAF</strong>";
echo $this->GTNA_Security->strip_tag($tag,"<strong>");
```

به این شکل، فقط تگ strong بر روی کلمه GTNAF اجرا می‌شود. برای اینکه بهتر متوجه شوید، پس از اجرای کد بالا توسط مرورگر سورس آن را مشاهده کنید.

اگر می‌خواهید هر دو تگ اجرا شود، یعنی تگ p و تگ strong به شکل زیر عمل کنید:

```
$tag="<p>this is a test</p><strong>GTNAF</strong>";
echo $this->GTNA_Security->strip_tag($tag,"<strong><p>");
```

متد **escape()** همانطور که از نامش پیداست وظیفه معتبرسازی داده‌ها را دارد. بیشترین کاربرد آن در ارسال فرم‌ها و کار با متد GET, POST است، چون امکان دارد کاربران داده‌های نامعتبر به فرم ارسال کنند. باید آنها را به لحاظ داده‌ای مجاز کنیم و سپس داده‌ها را برای ثبت به لایه دیتا ارسال کنیم و در مرحله آخر عمل درج را انجام دهیم.

```
$data="<script>alert('this is a test');</script>";
echo $this->GTNA_Security->escape($data);
```

خروجی:

```
&#60;script&#62;alert(&#39;this is a test&#39;);&#60;/script&#62;
```



1.7.GFB فریم ورک گتتا نسخه

با مشاهده خروجی، می‌بینید که به دلیل اینکه داده غیرمجاز است، تمامی تگ‌ها به کدهای نامعتبر تبدیل شده‌اند. به این ترتیب حتی اگر این داده درون جدول ذخیره شود، به عنوان یک رشته به خروجی باز می‌گردد و هیچ خطری برای سایت ندارد. البته باید آنقدر دانش داشته باشید که بتوانید با جستجو درون رشته اسکریپت‌های خطرناک را پیدا کنید و در صورت وجود آنها، داده‌های فرم را به عنوان داده‌های نامعتبر تلقی کنید و پیام مربوطه را به کاربر نمایش دهید. حتی در صورتی که داده مجاز باشد به وسیله متد زیر داده‌ها را در لایه دیتا بررسی کنید.

db->Mysql_sanitize_string();

توجه: متد **db->Mysql_sanitize_string()** فقط در لایه یا داده‌ها و زمانی که به بانک اطلاعاتی متصل هستید عمل می‌کند، در غیر این صورت با پیامی مبنی بر اینکه شما به بانک اطلاعاتی ارتباط ندارید برخورد می‌کنید. برای آشنایی با طرز استفاده از این متد، به قطعه کد زیر توجه کنید.

```
$this->db->open();
$data="<script>alert('this is a test');</script>";
echo $this->db->Mysql_sanitize_string($data);
```

تذکر: تمام فیلدهایی که قرار است درون یک جدول ذخیره شوند، باید از این فیلتر عبور کنند.

متد **GTNA_Security->encrypt()**: وظیفه این متد رمز نگاری داده‌ها می‌باشد. بیشترین زمانی که به این متد نیاز پیدا می‌کنید، برای ذخیره‌سازی پسورد کاربران درون بانک اطلاعاتی می‌باشد. قطعه کد زیر نحوه استفاده از این متد را بیان می‌کند.

```
$pass="123"
echo $this->GTNA_Security->encrypt($pass);
```

در این قسمت مثال‌هایی در مورد عملیات CRUD برای شما آورده‌ایم تا راحت‌تر بتوانید عملیات‌هایی مانند

درج، حذف، ویرایش و همچنین نحوه خواندن داده‌ها از درون جداول می‌پردازیم.

جدول زیر تمامی متدهایی می‌باشد که در لایه دیتا و یا همان مدل قابل استفاده می‌باشد:

db->Mysql_delete(table,query_string)	db->Mysql_open()
db->Mysql_update(table,array())	db->Mysql_close()
db->Mysql_select(table,query_string)	db->Mysql_insert(table,array)
db->Mysql_sanitiz_string(value)	



توجه: دستورات نسبت به حروف کوچک و بزرگ حساس است یعنی بین Mysql و Mysql تفاوت قائل می شود.

ابتدا در مسیر زیر تنظیمات مربوط به بانک اطلاعاتی را انجام دهید

application/config/database.php

۱- `db->Mysql_open()`: برای اتصال به بانک اطلاعاتی استفاده می شود.

هرزمان که قصد دارید در یک جدول عملیات های درج، حذف، و یا ویرایش انجام دهید قبل از آن باید کد زیر را بنویسید.

```
$this->db->Mysql_open();
```

۲- `db->Mysql_close()`: بعد از انجام هر یک از عملیات های CRUD به وسیله این متد حتماً اتصال با بانک اطلاعاتی را ببندید.

```
$this->db->Mysql_close();
```

۳- `db->Mysql_insert()`: برای درج درون جدول استفاده می شود. قطعه کد زیر مثالی در این رابطه می باشد.

فرض کنید جدولی با نام user به شکل زیر درون پایگاه داده ایجاد کرده ایم.

```
$user_name="gtna";
$user_pass=$this->db->mysql_sanitize_encrypt("123");
$email="admin@gtna.net";
$data=array(
    'user_name'=>$user_name,
    'user_pass'=>$user_pass,
    'email'=>$email);
$this->db->Mysql_open();
$this->db->Mysql_insert("user",$data);
$this->db->Mysql_close();
```

اگر به اندیس های آرایه \$data وفیلد های جدول نگاه کنید، می بینید که دقیقاً مانند یکدیگرند. بنابراین این نکته را در هنگام درج داده ها فراموش نکنید، در غیر این صورت با خطا برخورد می کنید.



۴- متد `Mysql_update(table,array())`: طبیعتاً از این متد برای به‌روزرسانی داده‌ها استفاده می‌شود. برای درک بهتر نحوه عملکرد این متد به مثال زیر توجه نمایید.

فرض کنید id شماره ۱۲ مربوط به رکوردی است که اکنون قصد به‌روزرسانی آن را داریم.

```
function update($data=NULL){
    $user_name="gtna_user"; // تغییر داده
    $user_pass=$this->db->mysql_sanitize_encrypt("456");// تغییر داده
    $email="gtna@att.net";// تغییر داده
    $data=array(
        'user_name'=>$user_name,
        'user_pass'=>$user_pass,
        'email'=>$email);
    $this->db->Mysql_open();
    $this->db->Mysql_update("user",$data,"12"); // به روز رسانی توسط آی دی کاربر
    $this->db->Mysql_close();
}
```

۵- `Mysql_select(table,query_string)`: به وسیله این متد عملیات مربوط به واکنشی داده‌ها انجام می‌شود به قطعه کد زیر توجه فرمایید.

```
function select(){
    $this->db->open();
    $data=$this->db->Mysql_select('user',"id=12");// خروجی از نوع آبجکت می باشد
    return $data;
}
```

Application/models/**select_model.php**

حالا باید به وسیله کنترلر داده‌ها را دریافت کنیم:

```
$this->load->model('select_model');
$this->library('select_model');
$result['data']= $this->select_model->select();
$this->load->view('dislay',$result);
```

Application/views/**dislay.php**



نوبت به نوشتن کدهای مربوط به view برای نمایش داده ها می‌رسد.

Display.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>display view</title>
</head>
<body>
```

```
<?php
    foreach ($data as $data)
        for($i=0;$i<$data['numrow'];$i++):
            echo $data[$i]->id."<br>";
            echo $data[$i]->user_name."<br>";
            echo $data[$i]->email."<br>";
            echo "_____<br>";
        endfor
?>
</body>
</html>
```

۶- متد **Mysql_delete(table,query_string)**: به وسیله این متد می‌توانید عملیات حذف را انجام دهید. برای آشنایی با نحوه استفاده از آن، مثالی ساده آورده‌ایم.

```
function delete(){
    $this->db->Mysql_open();
    If($this->db->Mysql_delete("user","id=12"))
        return TRUE;
    else
        return FALSE;
}
```



یک **view** با نام **form** در مسیر زیر ایجاد کنید.

Application/views/form.php

کدهای مربوط به فرم را به صورت زیر به یک متد خاص از کنترلر ارسال می‌کنیم. لطفاً به مثال زیر توجه کنید.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>display view</title>
</head>
<body>
```

```
<form action="<?php echo SYSTEM_URL."index.php/welcome/form"; ?>" method="post">
<p><input type="text" name="user_name">
<p><input type="password" name="user_pass " /></p>
<p><input type="submit" value="login" /></p>
</form>
```

اگر به مقدار خاصیت **action** فرم توجه کنید، متوجه می‌شوید که طبق قانون URLها که قبلاً درمورد آنها توضیحات کامل را داده‌ایم. این فرم به وسیله متد **POST** به کنترلر **welcome** و متد **form** ارسال شده است. بنابراین اگر کد زیر را در متد **form** بنویسید و سپس فرم را با اطلاعات مربوطه ارسال کنید، یک آرایه از داده‌های ارسال شده از طریق فرم دریافت و نمایش داده می‌شود.

```
print_r($_POST);
```

به این ترتیب می‌توانید داده‌ها را درون کنترلر اعتبارسنجی کنید و سپس در صورت لزوم از آنها درمدل‌ها استفاده نمایید.



روش استفاده از کتابخانه GTNA_HTML

اگر به وسیله زبان پی اچ پی یک interface ساده طراحی کرده باشید می دانید که کد های HTML و PHP درهم می شوند و زمانی که بخواهید کدهای قبلی را ویرایش کنید کارتان بسیار دشوار می شود. بنابراین گتتا یک کتابخانه در فریم ورک تعبیه کرده است که تگهای HTML توسط متدهای آن پردازش می شود و پس از اجرا اشیاء به وجود می آیند. به جدول متد های این کتابخانه توجه نمایید.

<code>GTNA_HTML::select(array(property));</code>	<code>GTNA_HTML::input(array(peroperty));</code>
<code>GTNA_HTML::option_property(array(property),'name');</code>	<code>GTNA_HTML::img(array(property));</code>
<code>GTNA_HTML::option('value');</code>	<code>GTNA_HTML::textarea(array(property),value);</code>
<code>GTNA_HTML::end_select(void);</code>	<code>GTNA_HTML::form_open(array(property));</code>
<code>GTNA_HTML::a(array(property),linkname);</code>	<code>GTNA_HTML::form_close(void);</code>

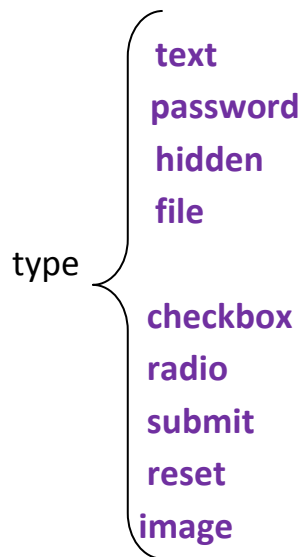
حتما با خود می گوئید با این چند متد چطور می توان یک فرم کامل را به وجود آورد؟

۱- `GTNA_HTML::input(array(peroperty))`: تمام اشیایی که با تگ `input` ایجاد می شوند توسط این متد پردازش و ایجاد می شود همانطور که می بینید یک آرایه از خصوصیات را دریافت می کند و به کد HTML تبدیل می کند.

توجه داشته باشید که تمام کدها را در لایه `view` بنویسید.

```
<?php//
    $text=array(
        'name'=>'user_name',
        'type'=>'text',
        'class'=>'txt');    // تمام خصوصیتی که لازم است را می توانید به آرایه بیافزایید.
?>
<?php echo GTNA_HTML::input($text); ?>
```

اگر درقطعه کد بالا درآرایه به فیلد `type` دقت کنید نوع شیء را مشخص می کند، این فیلد می تواند مقادیر زیر را بپذیرد.



بنابراین کافی است بجای مقدار فیلد **type** از مقادیر بالا استفاده کنید ولی ممکن است خاصیت های بعضی از اشیاء فرق داشته باشد کافی است فیلد جدید اضافه کنید و نام آن خاصیت را بنویسد .

۲- **GTNA_HTML::select(array(property))** : از این متد برای ایجاد **ComboBox** استفاده می شود اما زمانی که از این متد استفاده می کنید حتما باید از متد **GTNA_HTML::option('value');** و در پایان از متد **GTNA_HTML::end_select(void);** استفاده کنید اگر قصد دارید **option** ها را به همراه خاصیت های خاص ایجاد کنید باید از متد **GTNA_HTML::option_property(array(property));** استفاده کنید .

```
<?php//
    $select=array(
        'name'=>'user_name',
        'type'=>'select',
        'id'=>'select-style');
?>
<?php echo GTNA_HTML::select($select); ?>
    <?php echo GTNA_HTML::option('مرد'); ?>
    <?php echo GTNA_HTML::option('زن'); ?>
<?php echo GTNA_HTML::end_select(); ?>
```



به این شکل یک ComboBox ساختم اما اگر قصد دارید که option های خود را با استایل و خاصیت های مورد نظرتان ایجاد کنید به شکل زیر عمل کنید .

```
<?php//
$select=array(
    'name'=>'user_name',
    'type'=>'select',
    'id'=>'select-style');
$option_property=array(
    'id'=>'option-style',
    '...', '...'// می توانید خاصیت های دیگر را به عنوان فیلد به آرایه بیافزاید.
?>
<?php echo GTNA_HTML::select($select); ?>
<?php echo GTNA_HTML::option_property($option_property, 'مرد'); ?>
<?php echo GTNA_HTML::option_property($option_property, 'زن'); ?>
<?php echo GTNA_HTML::end_select(); ?>
```

۳- `GTNA_HTML::img(array(property))`: از این متد برای ایجاد تصویر استفاده می شود

```
<?php//
$image=array(
    'height'=>'400px',
    'width'=>'400px',
    'src'=>IMAGE_URL."/your-pic.jpg");
?>
<?php echo GTNA_HTML::img($image); ?>
```

می توانید از طریق کنترلر بعد از آپلود تصاویر مشخصات آن را به صورت یک آرایه به view ارسال کرد و سپس به وسیله متد `GTNA_HTML::img(array(property))` تصاویر را ایجاد کنید در چنین موقعیت هایی استفاده از این متد معنا دار می شود ولی به صورت معمول کاربرد خاصی ندارد و می توان از همان تگ `img` استفاده کنید.



۴- `GTNA_HTML::a(array(property),linkname)`: مشخص است که از این متد برای ایجاد لینک استفاده

می شود .

```
<?php
$link=array(
    'title'=>'my link',
    'href'=>'http://www.google.com'
);
?>
<?php echo GTNA_HTML::a($link,'ورود به گوگل'); ?>
```

۵- `GTNA_HTML::textarea(array(property),value)`: برای ایجاد textarea استفاده می شود به مثال

زیر توجه کنید.

```
<?php
$textarea=array(
    'cols'=>'40',
    'rows'=>'30',
    '...' // در صورت لزوم خاصیت های دیگر را اضافه کنید
);
?>
<?php echo GTNA_HTML::textarea($textarea,'این یک مثال است'); ?>
```

۶- `GTNA_HTML::form_open(array(property))`: برای ایجاد فرم استفاده می شود و در پایان این متد

باید متد `GTNA_HTML::form_close(void);` قرار گیرد.

```
<?php
$form=array(
    'action'=>SYSTEM_URL."index.php/controller/method",
    'method'=>'post',
    '...' // در صورت لزوم خاصیت های دیگر را اضافه کنید
);
?>
<?php echo GTNA_HTML::form_open($form,'این یک مثال است'); ?>
<?php echo GTNA_HTML::input(array('type'=>'text','name'=>'user_name')); ?>
<?php echo GTNA_HTML::input(array('type'=>'text','password'=>'password')); ?>
<?php echo GTNA_HTML::input(array('type'=>'submit','value'=>'ورود به سایت')); ?>
<?php echo GTNA_HTML::form_close(); ?>
```



فریم ورک گتتا نسخه 1.7.GFB

شاید این تفکر به ذهنتان خطور کند که چرا دستورات ساده تر نشده اند مثلاً برای ایجاد یک شیء مانند text از input استفاده نکنیم و با دستور جداگانه بدون مشخص نمودن type آن را ایجاد کنیم در جواب به این سوال می گوییم بله این امکان وجود داشت ولی به لحاظ SEO مشکل به وجود می آمد و پویایی یا بهتر است بگوییم آزادی از طراح گرفته می شد.

اگر به تمام اشیاء ایجاد شده دقت کنید می بینید که شیء چند انتخابی وجود ندارد برای ایجاد این شی به صورت زیر عمل کنید .

```
<?php
$select=array(
    'multiple'=> 'multiple',
    ); // هر صفتی دیگری می توانید بیافزایید
?>
<?php echo GTNA_HTML::select($select); ?>
<?php echo GTNA_HTML::option('مرد'); ?>
<?php echo GTNA_HTML::option('زن'); ?>
<?php echo GTNA_HTML::end_select(); ?>
```



اعتبار سنجی فرم ها به وسیله کتابخانه GTNA_Validate

GTNA_Validate->mobile()	GTNA_Validate->IN()	GTNA_Validate->IP()
GTNA_Validate->UN()	GTNA_Validate->IR()	GTNA_Validate->URL()
GTNA_Validate->EN()	GTNA_Validate->Email()	GTNA_Validate->YMD()
GTNA_Validate->IREN()	GTNA_Validate->MC()	GTNA_Validate->AD()

این کتابخانه دارای یازده متد جهت اعتبار سنجی داده‌های درون فرم و یا خارج از آن می‌باشد. همان طور که می‌دانید هکرها برای اینکه داده‌های خود را به صورت یک کد مخرب به سایت تزریق کنند از طریق فرم‌ها اقدام می‌نمایند. این کتابخانه جهت مقابله با این گونه کدها ایجاد شده است در زیر هر کدام از متدهای موجود را به ترتیب توضیح داده ایم.

توجه : خروجی تمامی متدهای بالا مقادیر TRUE یا FALSE می‌باشد.

۱- متد GTNA_Validate->mobile()

این متد دارای دو پارامتر شماره و پیش شماره می‌باشد که در زیر روش استفاده از آن بیان شده است .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index()
    {
        $this->library('GTNA_Validate');
        $mobile_prefix=array('0917','0913','0919','0936');
        $state=$this->GTNA_Validate->mobile
            ('09174734146',$mobile_prefix);
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```



1.7. فریم ورک گتتا نسخه 1.7.GFB

خروجی : چون پیشوند ۰۹۱۷ درون آرایه وجود دارد بنابراین این شماره یک شماره معتبر می باشد اما توجه داشته باشید که شما باید شماره موبایل را از طریق فرم به کنترلر ارسال کنیم و از معتبر بودن آن مطمئن شویم .

۲- متد GTNA_Validate->UN()

این متد مخفف Unlimit Number می باشد یعنی وظیفه این متد پذیرش شماره نامحدود می باشد و تعداد ارقام مورد بررسی قرار نمی گیرد .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->UN
            ('0947341423123123123123126');
```

```
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```



۳- متد GTNA_Validate->EN()

این متد مخفف English می باشد یعنی به وسیله این متد کاربر فقط اجازه دارد یک مقدار انگلیسی ارسال کند .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->EN
('Hello world');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```

خروجی: چون رشته Hello world انگلیسی می باشد بنابراین پیام this is valid به نمایش در می آید اما اگر شما قصد دارید فقط معتبر بود آن را چک کنید می توانید پس از بررسی داده های وارد شده مقدار برگشتی آن را بررسی کنید.

۴- متد GTNA_Validate->IN() این متد فقط مقادیر Integer را می پذیرد .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->IN
(257);
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```



خروجی : چون ۲۵۷ یک مقدار معتبر و از نوع integer می باشد بنابراین معتبر است .

۵- متد `IR()->GTNA_Validate`: بررسی می کند که آیا رشته وارد شده توسط کاربر فارسی می باشد یا خیر.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->IR
        ('فریم ورک گتتا نسخه چهارم');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";}
    }
```

۶- متد `Email()->GTNA_Validate`: وظیفه ی این متد اعتبار سنجی ایمیل می باشد.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->Email('admin@gtna.net');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```



۷- متد `GTNA_Validate->IP()` با استفاده از این متد می‌توانید معتبر بودن یک IP را بررسی کنید.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->IP('192.168.20.20');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```

۸- متد `GTNA_Validate->URL()` این متد آدرس‌های اینترنتی را بررسی می‌کند و در پایان یک مقدار TRUE یا FALSE بر می‌گرداند.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access
allowed");
class main extends GTNA_controller{
    function index()
    {
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->URL('www.gtna.net');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
    }
}
```



۹- متد GTNA_Validate->YMD()

این متد مخفف کلمات Year Mount Day می‌باشد. وظیفه این متد بررسی تاریخ شمسی با توجه به سال جاری است. یعنی بررسی می‌کند که سال تاریخ وارد شده با سال جاری که اکنون در آن هستیم یکسان است یا خیر.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->YMD('1392/12/21');
```

```
if($state)
    echo "this is valid";
else
    echo "this is not valid";
    }
}
```

۱۰- متد GTNA_Validate->AD(): این متد بدون در نظر گرفتن سال جاری تاریخ‌های شمسی و میلادی را بررسی می‌نماید.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
        $state=$this->GTNA_Validate->AD('1393/12/21');
        if($state)
            echo "this is valid";
        else
            echo "this is not valid";
        }
    }
```



۱۱- متد `MC()` `GTNA_Validate->`: به وسیله این متد می‌تواند کدهای ملی را اعتبارسنجی کنید.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_Validate');
```

```
$state=$this->GTNA_Validate->MC('2572626042');
    if($state)
        echo "this is valid";
    else
        echo "this is not valid";
    }
}
```

۱۲- `GTNA_Validate->IREN()`: ای متد داده‌های فارسی و انگلیسی را می‌پذیرد و در صورت معتبر بود مقدار `TRUE` ارسال می‌کند روش استفاده از آن مثل مثال قبل است فقط به جای `MC` از `IREN` استفاده کنید.

اعتبار سنجی حرفه ای فرم ها :

علاوه بر اعتبار سنجی به روش بالا گتتا روش دیگری را جهت اعتبار سنجی فرم‌ها در هسته فریم ورک قرار داده است پیشنهاد ما این است که از این روش برای اعتبار سنجی فرم‌های خود استفاده کنید .

فرض کنید در لایه `views` ویویی به نام `login_form` وجود دارد که دارای سه فیلد به صورت زیر می‌باشد .

```
<?php
$form_open=array(
    'action'=>SYSTEM_URL."index.php/main/check",
    'method'=>'post'
);
```



```
$fun=array(
    'type'=>'text',
    'name'=>'user_name',
    'value'=>$data['value']['user_name']
);

$fup1=array(
    'type'=>'password',
    'name'=>'UP1',
    'value'=>$data['value']['UP1']
);

$fup2=array(
    'type'=>'password',
    'name'=>'UP2',
    'value'=>$data['value']['UP2']
);

$fs=array(
    'type'=>'submit',
    'value'=>'login'
);

?>

<?php print_r($data['err']);?>

<?php echo GTNA_HTML::form_open($form_open);?>
<?php echo GTNA_HTML::input($fun);?>
<?php echo GTNA_HTML::input($fup1);?>
<?php echo GTNA_HTML::input($fup2);?>
<?php echo GTNA_HTML::input($fs);?>
<?php echo GTNA_HTML::form_close();?>
```



اگر به فرم بالا توجه داشته باشید می بینید که این فرم اطلاعات را به متد check از کنترلر main ارسال می کند .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index()
    {
        $this->load->view('login_form');
    }
    function check(){
        $this->library('GTNA_Validate');
        $msg=array(
            'you can used (a-zA-Z0-9_.) charecters.',
            'password is empty or XSS script',
            'confirm password is empty or XSS script',
            'not match password'
        );
        $r1=array('User');
        $r2=array('Pass');
        $r3=array('Pass');
        $rules=array($r1,$r2,$r3);
        $err=$this->GTNA_Validate->Control($_POST,$rules,$msg);
        if($_POST['UP1']!= $_POST['UP2'])
            @$err.= $msg[3]."<br>";
        $data['err']=$err;
        $data['value']=$_POST;
        if($err)
            $this->load->view('login_form',$data);
        else
            echo "form Submitted";

    } // END OF MAIN CONTROLLER
```



1.7.GFB فریم ورک گتتا نسخه

در متد **check** متغیری از نوع آرایه به نام **\$msg** تعریف شده است که تمام پیام های خطا به ترتیب عناصر فرم لاگین درونش قرار گرفته است یعنی اولین پیام مربوط به یوزر است و به همین ترتیب بقیه پیام ها را به فیلدهای بعدی تعلق دارد به این نکته توجه کنید چون عملکرد **GTNA_validate** بر طبق این اصول است .

بعد از تنظیم پیام ها نوبت به تنظیم قوانین یا **rule** می باشد که دوباره ترتیب اهمیت دارد یعنی ابتدا قوانین مربوط به فیلد یوزر سپس پسورد و در نهایت تکرار پسورد که حتما باید از نوع آرایه ای باشد زمانی که در آرایه متغیر **\$r1** می نویسید **User** یعنی هیچ کدی غیر از کد های معتبر قابل قبول نباشد و این را خود سیستم تشخیص می دهد شما می توانید یک فیلد را با چند نوع فیلتر اعتبار سنجی کنید که البته زیاد این اتفاق نمی افتد مثلا فرض کنید می خواهید کد ملی را با دو فیلتر اعتبار سنجی کنید یکی اینکه از نوع عددی باشد و دوم اینکه یک کد معتبر باشد برای این کار به شکل زیر عمل کنید فرض کنید **\$r2** عنصری بر روی فرم است که کد ملی توسط کاربر را دریافت می کند .

```
$r2=array('UN','MC');
```

این طبیعی است که هر فیلتر در صورت نا معتبر بودن یک پیام خطا داد چون **\$r2** دارای دو فیلتر است و **\$r1** قبل از آن وجود دارد بنابراین اندیس صفر به **\$r1** تعلق دارد شما باید اندیس های اول و دوم از متغیر **\$msg** پیام های مربوط به فیلترینگ **UN,MC** قرار دهید .

در نهایت اگر فرم معتبر نبود به صورت اتوماتیک خطاها در متغیر **\$err** قرار می گیرد که در صورتی که بخواهید مقدار وارد شده توسط کاربر و همچنین خطاها در فرم به نمایش در آید می توانید آنها را صورت زیر درون متغیر قرار دهید و سپس مقادیر را به ویوی **login_form** ارسال کنید

```
$data['err']=$err;
```

```
$data['value']=$_POST;
```

و اگر خطایی وجود نداشته باشد به معنی این است که فرم شما کاملا معتبر و بدون خطر است آنگاه می توانید عملیات مورد نظر خود را انجام دهید

توجه کنید که این نوع اعتبار سنجی فقط مختص به فرم ها نیست و شما می توانید در موارد دیگر از این روش استفاده کنید البته این حالت به ندرت پیش می آید که می توانید از اعتبار سنجی نوع اول استفاده کنید .

توجه : می توانید از تمام فیلترینگ های جدول صفحه ۴۱ استفاده نمایید .



طراحی ماژول برای فریم ورک گتتا ؟

جهت طراحی پروژه های عظیم نیاز به تقسیم بندی پروژه و بارگذاری آنها در قسمت های دیگر می باشد که نام آن را HMVC گذاشته اند . فریم ورک هایی که تا کنون ایجاد شده اند معمولاً به صورت پیش فرض دارای این معماری نمی باشند ولی فریم ورک گتتا جهت راحتی طراحان این معماری را به صورت پیش فرض در خود جای داده است .

برای طراحی ماژول در فریم ورک گتتا ابتدا به مسیر زیر وارد شوید

Application/modules

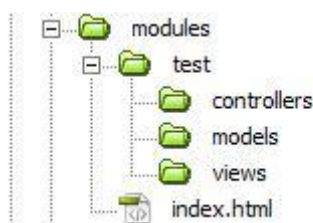
سپس در این مسیر دایرکتوری دیگری با نام ماژول دلخواه خود ایجاد کنید مثلاً test و دایرکتوری های زیر را در پوشه test ایجاد کنید.

controllers

models

views

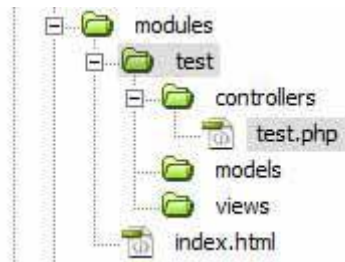
به شکل زیر دقت کنید



تنها نکته ای که باید در ایجاد کنترلر در نظر بگیرد این است که نام کنترلر موجود در دایرکتوری controllers باید با نام ماژول test یکی باشد به شکل زیر توجه کنید .



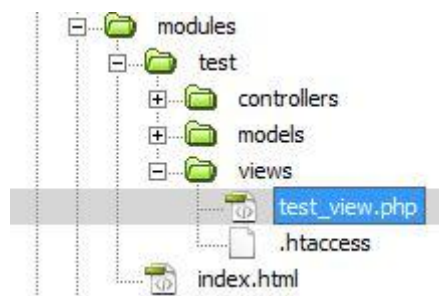
1.7.GFB فریم ورک گتتا نسخه



سپس کدهای کنترلر را مانند قبل ایجاد کنید فقط این نکته را توجه داشته باشید که نام کلاس کنترلر باید `test` باشد.

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class test extends GTNA_controller{
    function index{
        echo "this is test module";
    }
    Function MTV(){
        $this->load->view('test_view',NULL,'test');
    }
}
```

تذکر اگر در دایرکتوری `views` از ماژول `test` ویویی به شکل زیر ایجاد کرده اید نحوه دسترسی به ویو تغییر می کند.



برای بارگذاری ویو در کنترلر ماژول ها باید از کد زیر استفاده کنید .

```
$this->load->view('view name',data , 'module name');
```

`view name` : نام ویویی که تمایل دارید آن را لود کنید .



data: متغیر داده هایی که از بانک اطلاعاتی استخراج می شود را در این مکان قرار می دهند .

module name: نام ماژول که در اینجا همان **test** است.

اگر به کدهای متد MTV از کنترلر **test** دقت کنید متوجه می شوید که ویو چگونه فراخوانی شده است .

بیشتر اوقات یک ماژول حرفه ای نیاز به مدل دارد برای لود مدل ها در ماژولها باید مدل را به شکل زیر فراخوانی کنید .

```
$this->load->model('model name','module name');
```

در پایان پس از طراحی ماژول به دوروش می توانید به آن دسترسی داشته باشید .

۱- از طریق URL

۲- بارگذاری در ویو

جهت دسترسی به ماژول تست از طریق URL به صورت زیر عمل کنید .



اگر به URL توجه کنید کلمه **modules** که اجباری است و همیشه باید وجود داشته باشد بعد از آن نام ماژول طراحی شده است و بعد از آن یعنی **index** نام متد می باشد و از آن پس هر داده ای که ارسال کنید به عنوان مقادیر به متد ارسال می شود این روش زمانی مورد استفاده قرار می گیرد که ماژول شما بزرگ است مثلاً فرض کنید یک ماژول فروشگاه ساخته اید که همه جزئیات آن فقط درون خودش است اما گاهی یک ماژول جهت بارگذاری در یک قسمت سایت نوشته شده است که لازم می شود دورن ویو های اصلی سایت و در قسمت های مختلف بارگذاری شود آن گاه در ویو دلخواه به صورت زیر جهت بارگذاری ماژول اقدام کنید .

```
Load::module('module name','method',array(value1,value2));
```

توجه کنید که بیشتر از ۸ مقدار نمی توانید به متد ارسال کنید بنابراین جهت بارگذاری ماژول **test** در ویو به صورت زیر عمل کنید .

```
load::module('test','index',NULL);
```



ارسال داده های در ماژولها به متدهای کنترلرها

در بسیار مواقع نیاز است که داده های بدست آمده توسط ماژول به متد خاصی از کنترلر های اصلی ارسال شود و سپس به نمایش در آید برای این کار از روش زیر استفاده کنید .

```
<?php
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index(){
        $this->library('GTNA_URL');
        $this->GTNA_URL->SD('test','test',array(NULL,NULL,"GTNA"));
    }
}
```

متد SD یعنی Send Data برای ارسال داده ها به کنترلرها استفاده می شود زمانی که شما یک ماژول طراحی می کنید و آن را در قالب اصلی بارگذاری می کنید ممکن است تعدادی کوئری در کنترلری که قالب اصلی را نمایش می دهد اجرا شود بنابراین با بارگذاری ماژول کوئری های اصلی اجرا نمی شوند برای رفع این اشکال از متد SD استفاده کنید داده های را به کنترلر اصلی منتقل کنید و سپس همزمان با کوئری های دیگر در قالب اصلی نمایش دهید تا اشکالی در قالب بروز نکند.

متد SD سه آرگومان دارد که به سورت زیر است :

آرگومان اول: نام کنترلر اصلی است یعنی کنترلری که درون ماژولها نباشد .

آرگومان دوم : نام متدی از کنترلر اصلی می باشد که داده ها به آن ارسال می شود .

آرگومان سوم: از نوع آرایه ای است و داده های را به پارامترهای متد ارسال می کند .

اگر به کد بالا توجه کنید می بینید دو قسمت نال یا NULL دارد این به این معنی است که متد کنترلر اصلی دارای سه پارامتر بوده است که ما داده را به پارامتر سوم ارسال کردیم و هیچ مقداری به پارامتر اول و دوم ارسال نکردیم تا داده های اصلی خراب نشوند و سپس از طریق پارامتر سوم کنترلر اصلی داده را دریافت می کنید و هر پردازشی که دوست داشته باشیم بر روی آن انجام می دهیم .



روش طراحی آپلودر با کتابخانه GTNA_Upload

نوشتن آپلودر برای بسیاری از سایت‌ها یک امر الزامی محسوب می‌شود بسیاری از فریم ورک‌هایی که وجود دارند در آپلود فایل مشکلاتی به وجود می‌آورند که به آنها اشاره خواهیم کرد. اگر مطلع باشید آپلود نسبت به نوع و پسوند فایل حساس است برای مشخص کردن نوع فایل تولیدکنندگان نرم افزار از **Mime Type** استفاده می‌کنند هر فایل، مایم مخصوص خود را دارد این امر سبب می‌شود که هکر نتواند اکسپلویت خود را به پسوند های دروغین به سایت تزریق کنید اکسپلویت قطعه کدی می‌باشد که با تغییر در رفتار سایت باعث نفوذ پذیر شدن سایت می‌شود بنابراین بررسی پسوند فایل کافی نیست چرا که هکر ها کد های خود را با تغییر پسوند به سایتتان تزریق می‌کنند .

بسیاری از فریم ورک‌های موجود نسبت به مرورگر حساسیت نشان می‌دهند مثلاً زمانی که فایل های خاصی را مجاز به آپلود می‌کنید ممکن است که در مرورگر خاصی همه چیز به درستی عمل کند اما در فایر فاکس و یا مرورگر های دیگر فایل‌های غیر مجاز به راحتی آپلود می‌شوند این موضوع بسیاری از طراحان را به سطوح آورده است و دائم به دنبال مایم‌های سازگار با مرورگر می‌گردند اما فریم ورک گتتا با فیلتر گذاری فایل‌ها با تمامی مرورگر ها سازگار است و به راحتی و در کمتر از ۵ دقیقه می‌توانید آپلودر خود را در چند سطر به اتمام برسانید و از امنیت آن آسوده خاطر باشید .

مراحل طراحی فایل آپلودر به وسیله کتابخانه‌های GTNA_Upload و GTNA_Mime:

در فریم ورک گتتا برای پیدا کردن مایم نیازی به جستجو نیست این امر به وسیله متدی خاص در کتابخانه مایم امکان پذیر است شما به راحتی می‌توانید فایلی که مایم آن را نمی‌دانید انتخاب کنید و کتابخانه، مایم آن فایل را به شما برگشت می‌دهد .

۱- طراحی ویو:

نکته ای که در طراحی ویو باید در نظر داشته باشید و گتتا نسبت به آن حساسیت نشان می‌دهد نام گذاری تگ **input** است نام این تگ باید **file** باشد و هیچ نام دیگری نمی‌پذیرد . بنابراین برای طراحی ویو آپلودر در مسیر ویوهای فایلی با نام **upload.php** ایجاد کنید.



Application/views/**upload.php**

```
<form enctype="multipart/form-data" action="
<?php echo SYSTEM_URL.'index.php/main/get_file'?>" method="POST">
  <!-- Name of input element determines name in $_FILES array -->
  Send this file: <input name="file" type="file" />
  <input type="submit" value="Send File" />
</form>
```

۲- طراحی کنترلر:

ابتدا در کنترلر main و متد index ویوی آپلود را فراخوانی می‌کنیم اگر به اکشن فرم دقت کنید متدی دیگری با نام get_file در کنترلر main وجود دارد که عملیات آپلود را انجام می‌دهد.

```
<?php
#email:admin@gtna.net
#support:http://www.gtna.net
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index()
    {
        $this->load->view('upload');
    }
    function get_file(){
        $this->library('GTNA_Mime');
        $this->library('GTNA_Upload');
        $ext='zip,rar,jpg,mp3,rtf';
        $type=$this->GTNA_Mime->Type($ext);
        $state=$this->GTNA_Upload->MTC($_FILES,$type,$ext,'10');
```



```

if($state)
{
    $data=$this->GTNA_Upload->Save($_FILES,"application/uploads/");
    if(is_array($data)){
        echo "uploaded";
    }else
        echo "can not upload";
    }else
        echo "not valid";
} // End state if
} // End get_file method

```

ابتدا در متد `get_file()` دو کتابخانه `GTNA_Mime` , `GTNA_Upload` فراخوانی شده است کتابخانه-
ی مایم به وسیله پسوندهای مجاز که توسط طراح تعریف شده است مایم فایل ها را به صورت آرایه بر می
گرداند و کتابخانه آپلود به لحاظ امنیتی ابتدا نوع حقیقی فایل را بر می گرداند و سپس نام فایل را بررسی
می کند اگر کارکترهای غیر مجاز درون نام فایل باشد آنها را حذف می کند توجه داشته باشید اگر نام فایل
از مجموعه ای از کارکترهای غیر مجاز تشکیل شده باشد نام به صورت کامل حذف می شود و پیامی مبنی
بر اینکه نام فایل شما غیر مجاز است به نمایش در می آید.

در متغیر `$ext` به معنی اکستنشن پسوند های دلخواه خود را تعریف کنید اگر نمی دانید از چه پسوندهایی
استفاده کنید در پایان لیست کاملی از آنها به همراه مایم آمده است .

متد `Type` از کتابخانه مایم به وسیله پسوند ها لیست مایم های مجاز را به دست می آورد و سپس به
وسیله ی متد `MTC مخفف Mime Type Ckeck` فایل ارسالی توسط فرم را بررسی می کنیم تا خیر مجاز
نباشد این متد پنج پارامتر می پذیرد

- ۱- `$_FILES` : اطلاعاتی مربوط به فایلی که توسط فرم آپلود ارسال شده است درونش قرار دارد.
- ۲- `$type` یا آرگومان دوم : این متغیر آرایه ای از مایم های مجاز را درون خودش قرار می دهد.
- ۳- `$ext` یا آرگومان سوم: لیستی از پسوند های مجاز را درون خودش قرار می دهد در فریم ورک گتتا
جهت آپلود فایل علاوه بر مایم تایپ پسوند فایل ها بررسی می شود .



۴- آرگومان چهارم : نشان دهنده حجم مجاز برای آپلود فایل می باشد اگر به قطه کد بالا توجه کنید مقدار ۱۰ وارد شده است یعنی فایل هایی با حجم ۱۰ مگابایت و یا کمتر قابلیت آپلود داشته باشند بنابراین به این نتیجه می رسیم که آرگومان چهارم حجم بر اساس مگابایت است .

۵- آرگومان پنجم : این آرگومان مقدار ثابت RMT را می پذیرد و مخفف Return Mime Type است زمانی آرگومان پنجم استفاده می شود که طراح مایم فایل را نمی داند.

```
echo $this->GTNA_Upload->MTC($_FILES,$type,$ext,'10','RMT');
```

تذکر: زمانی که آرگومان پنجم استفاده می شود آپلودر گتنا به اشکال برخورد می کند بنابراین بعد از اینکه مایم فایل را بدست آوردید آرگومان پنجم را خالی بگذارید .

متد Save از کتابخانه آپلود دو آرگومان دارد :

۱- مشخصات فایل ارسالی توسط فرم آپلود

۲- مسیری که فایل پس از آپلود در آن قرار می گیرد

این متد آرایه ای از مشخصات فایل آپلود شده برگشت می دهد بنابراین مقدار برگشتی را با تابع is_array بررسی کنید.

مقادیر برگشتی این تابع به صورت زیر است

۱- Relative_url: مقدار این اندیس در واقع مسیر نسبی فایل است .

۲- Logical_url : مسیر منطقی فایل آپلود شده است .

۳- Size : حجم فایل آپلود شده .

به وسیله مقادیر برگشتی لینک فایل بر روی سرور به همراه حجم در اختیار طراح قرار می گرد تا طراح بتواند با استفاده از مشخصات عملیات مورد نظر خود را انجام دهد .

توجه : اگر فایل مجاز به آپلود نباشد با پیام No Valid روبرو می شوید و اگر فایل آپلود شود آرایه ای به همراه مشخصات فایل آپلود شده در متغیر دلخواه شما قرار می گیرد و پیام Uploaded نمایش داده می شود و اگر به هر طریقی همه مشخصات ارسال شده مجاز باشد ولی عملیات آپلود بر اثر اختلال در سیستم و یا سرور انجام نشود پیام can not upload نمایش داده می شود. البته این پیام ها قابل تغییر است.

لیست از پسوند ها و مایم های مجاز برای استفاده از فایل آپلودر در فریم ورک گتنا .



1.7. فریم ورک گتتا نسخه 1.7.GFB

رنگهای بنفش پسوندها و مایم های هستند که در آپلودر گتتا مجاز به استفاده می باشند.

text/plain Plain (c php c++)	Plain text. Default if data is primarily text and no other type detected.
text/html html	HTML. Default if common tags detected and server does not supply image/* type.
text/xml xml	XML data. Default if data specifies <?xml with an unrecognized DTD.
text/richtext rtf	Rich Text Format (RTF).
text/scriptlet scriptlet	Windows script component.
audio/x-aiff audio	Audio Interchange File, Macintosh.
audio/basic audio	Audio file, UNIX.
audio/mid mid	Windows Internet Explorer 7 and later. MIDI sequence.
audio/wav wav	Pulse Code Modulation (PCM) Wave audio, Windows.
image/gif gif	Graphics Interchange Format (GIF).
image/jpeg jpg	JPEG image.
image/pjpeg jpg	Default type for JPEG images.
image/png png	Internet Explorer 7 and later. Portable Network Graphics (PNG).
image/x-png png	Internet Explorer 7 and later. Default type for PNG images.
image/tiff tiff	Tagged Image File Format (TIFF) image.
image/bmp bmp	Bitmap (BMP) image.
image/x-bitmap bitmap	Removed from Windows Internet Explorer 8.
image/x-jg jg	AOL Johnson-Grace compressed file.
image/x-emf emf	Enhanced Metafile (EMF).
image/x-wmf wmf	Windows Metafile Format (WMF).
video/avi avi	Audio-Video Interleaved (AVI) file.
video/mpeg mpeg	MPEG stream file.
application/octet-stream bf	Binary file. Default if data is primarily binary.
application/postscript postscript	PostScript (.ai, .eps, or .ps) file.
application/base64 base64	Base64-encoded bytes.
application/macbinhex40 macbinhex40	BinHex for Macintosh.
application/pdf pdf	Portable Document Format (PDF).
application/xml xml	XML data. Must be server-supplied. See also "text/xml" type.
application/atom+xml atom	Internet Explorer 7 and later. Atom Syndication Format feed.
application/rss+xml rss	Internet Explorer 7 and later. Really Simple Syndication (RSS) feed.
application/x-compressed tar	UNIX tar file, Gzipped.
application/x-zip-compressed zip	Compressed archive file.
application/x-gzip-compressed gzip	Gzip compressed archive file.
application/java java	Java applet.
application/x-rar-compressed rar	Rar file
application/x-7z-compressed 7z	7z-compressed



1.7.GFB فریم ورک گتنا نسخه

audio/x-mp3 mp3	Mp3 file
application/x-shockwave-flash swf	Flash file
application/x-msdownload exe	Executable (.exe or .dll) file

.doc	application/msword
.dot	application/msword
.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document
.dotx	application/vnd.openxmlformats-officedocument.wordprocessingml.template
.docm	application/vnd.ms-word.document.macroEnabled.12
.dotm	application/vnd.ms-word.template.macroEnabled.12
.xls	application/vnd.ms-excel
.xlt	application/vnd.ms-excel
.xla	application/vnd.ms-excel
.xlsx	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
.xltx	application/vnd.openxmlformats-officedocument.spreadsheetml.template
.xslm	application/vnd.ms-excel.sheet.macroEnabled.12
.xltn	application/vnd.ms-excel.template.macroEnabled.12
.xlam	application/vnd.ms-excel.addin.macroEnabled.12
.xlsb	application/vnd.ms-excel.sheet.binary.macroEnabled.12
.ppt	application/vnd.ms-powerpoint
.pot	application/vnd.ms-powerpoint
.pps	application/vnd.ms-powerpoint
.ppa	application/vnd.ms-powerpoint
.pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation
.potx	application/vnd.openxmlformats-officedocument.presentationml.template
.ppsx	application/vnd.openxmlformats-officedocument.presentationml.slideshow
.ppam	application/vnd.ms-powerpoint.addin.macroEnabled.12
.pptm	application/vnd.ms-powerpoint.presentation.macroEnabled.12
.potm	application/vnd.ms-powerpoint.template.macroEnabled.12
.ppsm	application/vnd.ms-powerpoint.slideshow.macroEnabled.12

اگر پسوند دلخواهتان در لیست وجود ندارد ابتدا مایم آن را به روش که گفتیم بدست بیاورد و آن را به فایل `gtna_mime.php` در شاخه `Config` اضافه کنید .

اگر یک مایم دارید آن را با # به اتمام برسانید اگر چندین مایم برای یک فایل وجود دارد فقط آخرین مایم را با # به اتمام برسانید و بقیه را با علامت @ خاتمه دهید .



افزایش سرعت فریم ورک گتتا

در ورژن هفت فریم ورک گتتا می توانید قسمت هایی از هسته که در پروژه شما مورد استفاده نیست را غیر فعال کنید تا قسمت های اضافی هسته بارگذاری نشود این امر در هیچ فریم ورکی رعایت نشده است اما گتتا با در نظر گرفتن نیاز کاربران و بنا به درخواست بسیاری از طراحان این موضوع را رعایت کرده است و شما از این به بعد قادر هستید که قسمت های غیر ضروری را غیر فعال کنید .

روش غیر فعال کردن کتابخانه های هسته :

جهت غیر فعال کردن کتابخانه های به مسیر زیر بروید

Application/config/GTNA-config.php

سپس می توانید متغیرهای زیر را در تنظیمات مشاهده کنید .

//=====Library config=====

```
$_CORE['MIME']          =    'ON';
$_CORE['CAPTCHA']       =    'OFF';
$_CORE['DATABASE']      =    'OFF';
$_CORE['HTML']          =    'OFF';
$_CORE['JAVA']          =    'OFF';
$_CORE['LIB_LOADER']    =    'OFF';
$_CORE['SESSION']       =    'OFF';
$_CORE['VALIDATOR']     =    'OFF';
$_CORE['SECURITY']      =    'OFF';
$_CORE['MODEL']         =    'OFF';
$_CORE['URL']           =    'OFF';
$_CORE['UPLOAD']        =    'ON';
$_CORE['CACHE']         =    'OFF';
```

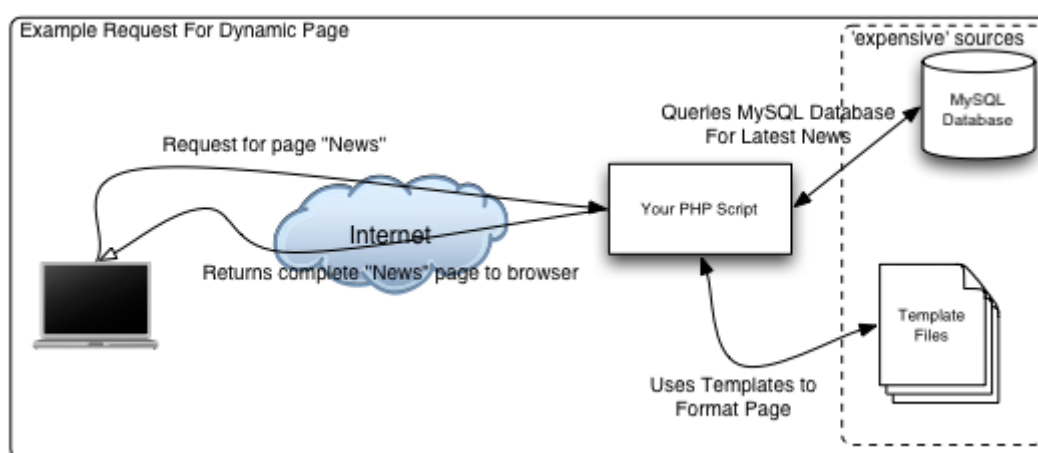


`$_CORE['ECARD'] = 'OFF';`

فکر نمی‌کنم نیازی به توضیح داشته باشد کافی است هر قسمت را که نیاز ندارید با کلمه OFF غیر فعال کنید و در غیر این صورت آنها باید ON باشند توجه کنید که نسبت به حروف کوچک و بزرگ حساس است.

سیستم کشینگ به وسیله GTNA_Cache:

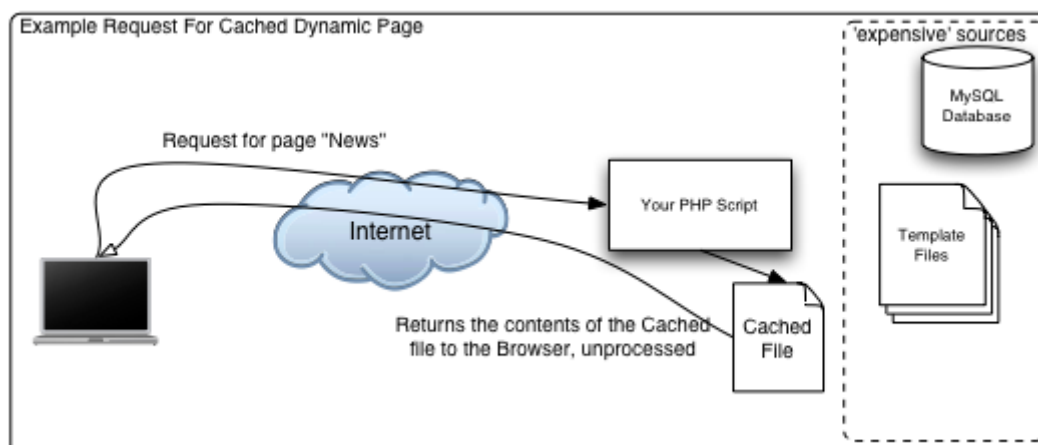
در ابتدا به شرح سیستم کش می‌پردازیم لطفاً به تصویر زیر نگاه کنید



با توجه به تصویر بالا مشاهده می‌کنید که هر زمان کاربر درخواستی را ارسال می‌کند اسکریپت‌های پی‌اچ‌پی آنها را به قسمت‌های متفاوت برای بارگذاری صفحه ارسال می‌کند و طی یک عملیات‌های متفاوت پردازشی توسط پی‌اچ‌پی جواب درخواست‌ها برگشت داده می‌شود و اگر این موضوع بیش از یک بار اتفاق بیافتد دوباره همین روال تکرار می‌شود و این یعنی سیستم همیشه در حال پردازش است. بنابراین چون همیشه سیستم در حال پردازش است سرعت بارگذاری صفحات را بسیار کند می‌کند با توجه به مطلب گفته شده تصویر بالا از هیچ سیستم کشینگ استفاده نکرده است البته توجه داشته باشید که اگر محتوای سایت شما دائم در حال تغییر است سیستم کشینگ مفید واقع می‌شود و در غیر این صورت تاثیری در سرعت بارگذاری صفحات ندارد.



کش چگونه سرعت را افزایش می دهد؟



در تصویر بالا مشاهده می کنید که زمانی که درخواست برای بار اول ارسال می شود خواه ناخواه باید پردازش صورت گیرد بعد از اولین پردازش فایلی از خروجی های درخواست های پردازش شده در مسیری از پروژه ایجاد می شود این مسیر می تواند دلخواه باشد حال اگر این درخواست برای بار دوم ارسال شود سیستم، فایل کش شده را فراخوانی می کند که نتیجه ی یک درخواست تکراری و پردازش شده است بنابراین نیازی به پردازش مجدد نیست اما توجه داشته باشید که برای سیستم کش باید زمان در نظر گرفته شود در غیر این صورت هر بار که درخواست ارسال می شود مطالب و یا پست های جدید بارگذاری نمی شوند و همیشه فایل کش شده قدیمی به نمایش در می آید بنابراین طراحان با توجه با نظر خودشان مدت زمانی برای سیستم کش در نظر می گیرند با اتمام آن مدت زمان دوباره درخواست، پردازش می شود و تا زمانی که شما برای سیستم مشخص کرده اید از فایل های کش استفاده می کند.



کش در فریم ورک گتنا :

سیستم کش در فریم ورک گتنا بسیار آسان است و نیازی به کدهای پیچیده ندارد به قطه کد زیر توجه کنید .

```
<?php
#email:admin@gtna.net
#support:http://www.gtna.net
if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");
class main extends GTNA_controller{
    function index()
    {
        $this->library('GTNA_Cache');
        $this->GTNA_Cache->Open();
        $this->load->view('welcome');
        $this->GTNA_Cache->Save('home',2*60);
    }
}
```

در ابتدا کتابخانه GTNA_Cache را فراخوانی می‌کنیم سپس به وسیله متد Open کش را باز کنید تا درخواست های پردازش شده در دسترس قرار گیرند سپس هر ویوی که دوست دارید کش شود را فراخوانی کنید و در مرحله آخر به وسیله متد Save ویوها را کش کنید متد Save دارای دو آرگومان جهت فراخوانی می‌باشد.

۱- نام فایل

۲- متد زمان کش

توجه : از نام های تکراری برای کش استفاده نکنید .

اگر به آرگومان دوم توجه کنید عدد دو در شصت ضرب شده است این به معنی دو دقیقه کش است و بعد از گذر از دو دقیقه سیستم دوباره ویوی welcome را به همراه تغییرات جدید به مدت دو دقیقه از اولین درخواست کش می‌کند.



توجه : هر ویوی که قصد کش کردن آن را دارید باید بین متد Open و Save قرار گیرد در غیر این صورت کش نمی شود.

نصب Nusoap بر روی فریم ورک گتتا

مفهوم web service , soap چیست؟

کسانی که با صنعت IT آشنایی دارند حتماً نام وب سرویس را شنیده اند. برای مثال، بیش از ۶۶ درصد کسانی که در نظر سنجی مجله InfoWorld شرکت کرده بودند بر این توافق داشتند که وب سرویس ها مدل تجاری بعدی اینترنت خواهند بود. به علاوه گروه گارتنر پیش بینی کرده است که وب سرویس ها کارآیی پروژه های IT را تا ۳۰ درصد بالا می برد. اما وب سرویس چیست و چگونه شکل تجارت را در اینترنت تغییر خواهد داد؟

برای ساده کردن پردازش های تجاری، برنامه های غیرمتمرکز (Enterprise) باید با یکدیگر ارتباط داشته باشند و از داده های اشتراکی یکدیگر استفاده کنند. قبلاً این کار به وسیله ابداع استانداردهای خصوصی و فرمت داده ها به شکل مورد نیاز هر برنامه انجام می شد. اما دنیای وب و XML تکنولوژی آزاد برای انتقال اطلاعات بین سیستم ها را افزایش داد. وب سرویس ها نرم افزارهایی هستند که از XML برای انتقال اطلاعات بین نرم افزارهای دیگر از طریق پروتکل های معمول اینترنتی استفاده می کنند. به شکل ساده یک وب سرویس از طریق وب اعمالی را انجام می دهد (توابع یا سابروتین ها) و نتایج را به برنامه دیگری می فرستد. این یعنی برنامه ای که در یک کامپیوتر در حال اجراست اطلاعاتی را به کامپیوتر دیگری می فرستد و از آن درخواست جواب می کند. برنامه ای که در آن کامپیوتر دوم است کارهای خواسته شده را انجام می دهد و نتیجه را بر روی ساختارهای اینترنتی به برنامه اول برمی گرداند.

وب سرویس ها می توانند از پروتکل های زیادی در اینترنت استفاده کنند اما بیشتر از HTTP که مهم ترین آنهاست استفاده می شود. به وسیله ای این تکنولوژی کارهای مختلفی صورت می گیرد. برای مثال در یک برنامه می تواند آخرین عنوان های اخبار را از وب سرویس Associated Press بگیرد یا یک برنامه مالی می تواند آخرین اخبار و اطلاعات بورس را از طریق وب سرویس بگیرد. کاری که وب سرویس انجام می دهد می تواند به سادگی ضرب دو عدد یا به پیچیدگی انجام کلیه امور مشترکین یک شرکت باشد.

وب سرویس دارای خواصی است که آن را از دیگر تکنولوژی ها و مدل های کامپیوتری جدا می کند. Paul Flessner، نایب رئیس مایکروسافت در dot NET Enterprise Server چندین مشخصه برای وب سرویس در یکی از نوشته هایش ذکر کرده است. اول اینکه وب سرویس ها قابل برنامه ریزی هستند. یک



وب سرویس کاری که می کند را در خود مخفی نگه می دارد. وقتی برنامه ای به آن اطلاعاتی می دهد وب سرویس آن را پردازش می کند و در جواب آن اطلاعاتی را به برنامه اصلی بر می گرداند. وب سرویس ها بر پایه XML بنا نهاده شده اند. XML و XML های مبتنی بر SOAP یا Object Access Simple Protocol تکنولوژی هایی هستند که به وب سرویس ها این امکان را می دهد که با دیگر برنامه ها ارتباط داشته باشد حتی اگر آن برنامه ها در زبانهای مختلف نوشته شده و بر روی سیستم عامل های مختلفی در تال اجرا باشند.

همچنین وب سرویس ها خود توصیف هستند. به این معنی که نحوه استفاده از خودشان را توصیف می کنند. این توضیحات به طور کلی در WSDL یا Description Language Web Services نوشته می شود. WSDL یک استاندارد بر مبنای XML است. به علاوه وب سرویس ها قابل شناسایی هستند به این معنی که برنامه نویس می تواند به دنبال وب سرویس مورد علاقه در دایرکتوری هایی مثل UDDI یا and Integration Universal Description , Discovery جستجو کند. UDDI یکی دیگر از استانداردهای وب سرویس است.

نکات تکنولوژی وب سرویس همانطور که در ابتدا توضیح داده شد یکی از دلایل اینکه وب سرویس از دیگر تکنولوژی های موجود مجزا شده است ، استفاده از XML و بعضی استاندارد های تکنیکی دیگر مانند SOAP، WSDL و UDDI است. این تکنولوژی ها زمینه ارتباط بین برنامه ها را به شکلی ایجاد می کنند که مستقل از زبان برنامه نویسی، سیستم عامل و سخت افزار است. SOAP یک مکانیزم ارتباطی را بین نرم افزار و وب سرویس ایجاد می کند. WSDL یک روش یکتا برای توصیف وب سرویس ایجاد می کند و UDDI یک دایرکتوری قابل جستجو برای وب سرویس می سازد. وقتی این تکنولوژی ها با هم در یک جا جمع می شوند به برنامه نویس اجازه می دهد که برنامه های خود را به عنوان سرویس آماده کرده و بر روی اینترنت قرار دهد.

XML یک تکنولوژی است که به شکل گسترده از آن پشتیبانی می شود، همچنین این تکنولوژی Open Source است به این معنی که متعلق به شرکت خاصی نیست. اولین بار در کنسرسیوم WWW یا W3C در سال ۱۹۹۶ برای ساده کردن انتقال دیتا ایجاد شده است. با گسترده شدن استفاده از وب در دهه ۹۰ کم کم محدودیت های HTML مشخص شد. ضعف HTML در توسعه پذیری (قابلیت اضافه و کم کردن خواص) و ضعف آن در توصیف داده هایی که درون خود نگهداری می کند برنامه نویسان را نا امید کرد. همچنین مبهم بودن تعاریف آن باعث شد از توسعه یافتن باز بماند. در پاسخ به این اشکالات W3C یک



سری امکانات را در جهت توسعه HTML به آن افزود که امکان تغییر ساختار متن‌های HTML مهم‌ترین آن است. این امکان را CSS یا Cascade Style Sheet می‌نامند.

این توسعه تنها یک راه موقت بود. باید یک روش استاندارد شده، توسعه پذیر و دارای ساختار قوی ایجاد می‌شد. در نتیجه W3C استاندارد XML را ساخت. XML دارای قدرت و توسعه پذیری SGML یا Standard Generalized Markup Language و سادگی که در ارتباط در وب به آن نیاز دارد است.

استقلال اطلاعات یا جدا بودن محتوا از ظاهر یک مشخصه برای XML به حساب می‌آید. متن‌های XML فقط یک داده را توصیف می‌کنند و برنامه‌ای که XML برای آن قابل درک است بدون توجه به زبان و سیستم عامل قادر است به اطلاعات درون فایل XML هر گونه شکلی که مایل است بدهد. متن‌های XML حاوی داده‌هایی بدون شکل خاص می‌باشد. بنابراین برنامه‌ای که از آن می‌خواهد استفاده کند باید بداند که چگونه می‌خواهد آن اطلاعات را نمایش دهد. در نتیجه نحوه نمایش یک فایل XML در یک PC با PDA و تلفن همراه می‌تواند متفاوت باشد.

وقتی یک برنامه با متن XML روبرو می‌شود باید مطمئن باشد که آن متن حاوی داده‌های مورد نظر خود است. این اطمینان توسط برنامه‌هایی با نام XML Parser حاصل می‌شود. تجزیه کننده‌ها دستورات متن XML را بررسی می‌کنند. همچنین آنها به برنامه کمک می‌کنند تا متن‌های XML را تفسیر کند. به صورت اختیاری هر متن XML می‌تواند به متن دیگری اشاره کند که حاوی ساختار فایل XML اصلی باشد. به آن متن XML دوم DTD یا Document Type Definition گفته می‌شود.

وقتی فایل XML به DTD اشاره می‌کند برنامه تجزیه کننده فایل اصلی را با DTD بررسی می‌کند که آیا به همان ساختاری که در DTD توصیف شده شکل گرفته است یا خیر. اگر یک تجزیه کننده XML بتواند یک متن را به درستی پردازش کند متن XML نیز به شکل صحیتی فرمت شده است.

وقتی که اکثر نرم افزارها امکانات تحت وب خود را افزایش دادند این طور به نظر می‌رسد که XML به عنوان یک تکنولوژی جهانی برای فرستادن اطلاعات بین برنامه‌ها انتخاب شود. تمامی برنامه‌هایی که از XML استفاده می‌کنند قادر خواهند بود که XML همدیگر را بفهمند. این سطح بالای تطابق بین برنامه‌ها باعث می‌شود که XML یک تکنولوژی مناسب برای وب سرویس باشد. چون بدون اینکه احتیاج به سیستم عامل و سخت افزار یکسان باشد می‌تواند اطلاعات را جابجا کند.



Simple Object Access Protocol SOAP

SOAP یکی از عمومی ترین استاندارد هایی است که در وب سرویس ها استفاده می شود. طبق شواهد اولین بار توسط DeveloperMentor، شرکت UserLand و مایکروسافت در سال ۱۹۹۸ ساخته شده و نسخه اول آن در سال ۱۹۹۹ ارایه شده است. آخرین نسخه SOAP، نسخه ۱.۲ بود که در دسامبر سال ۲۰۰۱ در W3C ارایه شد. نسخه ۱.۲ نشان دهنده کار زیاد بر روی آن و نمایانگر اشتیاق زیاد صنعت IT برای استفاده از SOAP و وب سرویس است.

هدف اصلی SOAP ایجاد روشی جهت فرستادن داده‌ها بین سیستم هایی است که بر روی شبکه پخش شده اند. وقتی یک برنامه شروع به ارتباط با وب سرویس می کند، پیغام های SOAP وسیله ای برای ارتباط و انتقال داده‌های بین آن دو هستند. یک پیغام SOAP به وب سرویس فرستاده می شود و یک تابع یا سابروتین را در آن به اجرا در می آورد به این معنی که این پیغام از وب سرویس تقاضای انجام کاری را دارد. وب سرویس نیز از محتوای پیغام SOAP استفاده کرده و عملیات خود را آغاز می کند. در انتها نیز نتایج را با یک پیغام SOAP دیگر به برنامه اصلی می فرستد.

به عنوان یک پروتکل مبتنی بر XML، پروتکل SOAP از الگوهای XMLی تشکیل شده است. این الگوها شکل پیغام‌های XML را که بر روی شبکه منتقل می شود را مشخص می کند. مانند نوع داده‌ها و اطلاعاتی که برای طرف مقابل تفسیر کردن متن را آسان کند. در اصل SOAP برای انتقال داده‌ها بر روی اینترنت و از طریق پروتکل HTTP طراحی شده است ولی از آن در دیگر مدلها مانند LAN نیز می توان استفاده کرد. وقتی که وب سرویس ها از HTTP استفاده می کنند به راحتی می توانند از Firewall عبور کنند.

یک پیغام SOAP از سه بخش مهم تشکیل شده است: پوشش یا Envelope، Header، بدنه یا Body. قسمت پوشش برای بسته بندی کردن کل پیغام به کار می‌رود. این بخش محتوای پیغام را توصیف و گیرنده آن را مشخص می کند. بخش بعدی پیغام های SOAP، Header آن است که یک بخش اختیاری می باشد و مطالبی مانند امنیت و مسیریابی را توصیف می‌دهد. بدنه پیغام SOAP بخشی است که داده‌های مورد نظر در آن جای می گیرند. داده‌ها بر مبنای XML هستند و از یک مدل خاص که الگوها (Schemas) آنها را توضیح می دهند تبعیت می‌کنند. این الگوها به گیرنده کمک می‌کنند تا متن را به درستی تفسیر کند. پیغام های SOAP توسط سرورهای SOAP گرفته و تفسیر می‌شود تا در نتیجه آن، وب سرویس‌ها فعال شوند و کار خود را انجام دهند.



برای اینکه از SOAP در وب سرویس استفاده نکنیم از تعداد زیادی پروتکل باید استفاده شود. برای مثال XML-RPC تکنولوژی قدیمی تری بود که همین امکانات را ایجاد می کرد. به هر حال، خیلی از سازندگان بزرگ نرم افزار SOAP را بر تکنولوژی های دیگر ترجیح دادند. دلایل زیادی برای انتخاب SOAP وجود دارد که خیلی از آنها درباره پروتکل آن است که فراتر از این متن می باشد. سه برتری مهم SOAP نسبت به یک تکنولوژی های دیگر عبارتند از قابلیت توسعه، سادگی و قابلیت عملکرد داخلی پیغام های SOAP معمولاً کدهای زیادی ندارند و برای فرستادن و گرفتن آن به نرم افزارهای پیچیده نیاز نیست. SOAP این امکان را به برنامه نویس می دهد تا بنا به نیاز خود آن را تغییر دهد. در آخر بدلیل اینکه SOAP از XML استفاده می کند می تواند بوسیله HTTP اطلاعات را انتقال بدهد بدون اینکه زبان برنامه نویسی، سیستم عامل و سخت افزار برای آن مهم باشد.

WSDL Language Web Services Description

استاندارد دیگری که نقش اساسی در وب سرویس بازی می کند WSDL است. همانطور که قبلاً اشاره کردیم یکی از خواص وب سرویس ها توصیف خود آنهاست به این معنی که وب سرویس دارای اطلاعاتی است که نحوه استفاده از آن را توضیح می دهد. این توضیحات در WSDL نوشته می شود، متنی به XML که به برنامه ها می گوید این وب سرویس چه اطلاعاتی لازم دارد و چه اطلاعاتی را بر می گرداند.

وقتی که سازندگان نرم افزار برای اولین بار SOAP و دیگر تکنولوژی های وب سرویس را ساختند دریافتند که برنامه ها قبل از اینکه شروع به استفاده از یک وب سرویس بکنند باید اطلاعاتی درباره آن را داشته باشند. اما هر کدام از آن سازندگان برای خودشان روشی برای ایجاد این توضیحات ابداع کردند و باعث شد که وب سرویس ها با هم هماهنگ نباشد. وقتی IBM و مایکروسافت تصمیم گرفتند تا استاندارد های خود را یکسان کنند WSDL بوجود آمد. در ماه مارس سال ۲۰۰۱ مایکروسافت، IBM و Ariba نسخه ۱.۱ را به W3C ارائه کردند. گروهی از W3C بر روی این استاندارد کار کردند و آن را پذیرفتند. هم اکنون این تکنولوژی در دست ساخت است و هنوز کامل نشده. ولی هم اکنون اکثر سازندگان وب سرویس از آن استفاده می کنند.

هر وب سرویسی که بر روی اینترنت قرار می گیرد دارای یک فایل WSDL است که مشخصات، مکان و نحوه استفاده از وب سرویس را توضیح می دهد. یک فایل WSDL نوع پیغام هایی که وب سرویس می فرستد و می گیرد را توضیح می دهد مانند پارامترهایی که برنامه صدا زننده برای کار با وب سرویس باید به آن بفرستد. در تئوری یک برنامه در وب برای یافتن وب سرویس مورد نظر خود از روی توضیحات WSDL



ها جستجو می کند. در WSDL اطلاعات مربوط به چگونگی ارتباط با وب سرویس بر روی HTTP یا هر پروتکل دیگر نیز وجود دارد.

این مهم است که بدانیم WSDL برای برنامه ها طراحی شده است نه برای خواندن آن توسط انسان. شکل فایل های WSDL پیچیده به نظر می آید ولی کامپیوترها می توانند آن را بخوانند و تجزیه و تحلیل کنند. خیلی از نرم افزارهایی که وب سرویس می سازند فایل WSDL مورد نیاز وب سرویس را نیز تولید می کنند بنابراین وقتی برنامه نویس وب سرویس خود را ساخت به شکل خودکار WSDL مورد نیاز با آن نیز ساخته می شود و احتیاجی به آموزش دستورات WSDL برای ساختن و استفاده از وب سرویس نیست.

Universal Description , Discovery and Integration ، UDDI

سومین استاندارد اصلی وب سرویس ها، یعنی UDDI، به شرکتها و برنامه نویسان اجازه می دهد تا وب سرویس های خود را بر روی اینترنت معرفی کنند. این استاندارد در اصل بوسیله مایکروسافت، IBM و Ariba و پنجاه شرکت بزرگ دیگر ساخته شده است. با استفاده از UDDI شرکتها می توانند اطلاعات خود را در اختیار شرکت های دیگر قرار بدهند و مدل B2B ایجاد کنند. همان طور که از نام آن مشخص است شرکت ها می توانند وب سرویس خود را معرفی کنند، با وب سرویس دیگران آشنا شوند و از آن در سیستم های خود استفاده کنند. این استاندارد جدیدی است و در سال ۲۰۰۰ ساخته شده است و کنسرسیومی از شرکتهای صنعتی در حال کار بر روی آن هستند. نسخه دوم UDDI در ماه ژوئن سال ۲۰۰۱ ارائه شد و نسخه سوم آن در دست ساخت است.

UDDI یک متن مبتنی بر XML را تعریف می کند که در آن شرکت ها توضیحاتی درباره چگونگی کار وب سرویس شرکتشان و امکانات خود می دهند. برای تعریف این اطلاعات از شکل خاصی که در UDDI توضیح داده شده استفاده می شود. شرکتها می توانند این اطلاعات را در UDDI شرکت خود نگهداری کنند و تنها به شرکت های مورد نظرشان اجازه دستیابی به آنها را بدهند یا آنها را در مکان عمومی و در اینترنت قرار دهند.

بزرگترین و مهمترین پایگاه UDDI پایگاه UDDI Business Registry یا UBR نام دارد و توسط کمیته UDDI طراحی و اجرا شده است. اطلاعات این پایگاه در چهار نقطه نگهداری می شود: مایکروسافت، IBM، SAP و HP. اطلاعاتی که در یکی از چهار پایگاه تغییر کند در سه تای دیگر نیز اعمال می شود.



1.7. فریم ورک گتتا نسخه 1.7.GFB

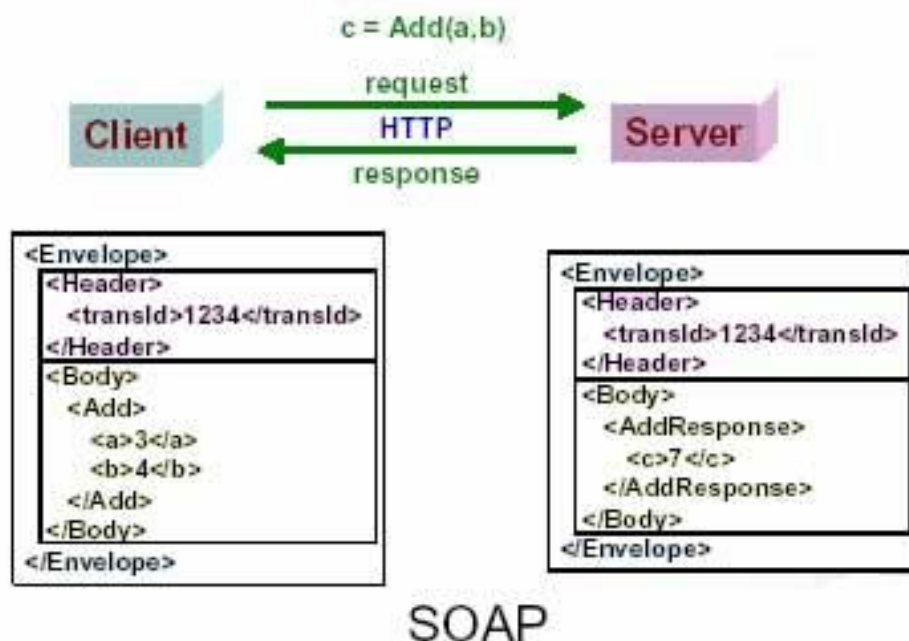
اطلاعات درون این پایگاه ها شبیه دفترچه تلفن است. **White Pages** که در آنها اطلاعات تماس شرکت ها و توضیحات متنی آنهاست، **Yellow Pages** حاوی اطلاعات طبقه بندی شده شرکتها و اطلاعات درباره توانایی های الکترونیکی آنها می باشد، **Green Pages**، حاوی اطلاعات تکنیکی درباره سرویس های آنها و نحوه پردازش اطلاعات شرکت آنها است.

اطلاعات تجاری و سرویس های شرکت ها کاملاً طبقه بندی شده است و اجازه می دهد که به راحتی در آنها جستجو انجام شود. سپس متخصصان **IT** می توانند از این اطلاعات استفاده کرده و شرکت ها را برای خدمات بهتر به هم متصل کنند. با این شرت **UDDI** امکان پیاده سازی مدل **B2B** را ایجاد می کند و شرکتها می توانند از سرویس های یکدیگر استفاده کنند.

شرکت هایی که به **UDDI** علاقه نشان داده اند قدرتمند هستند و خیلی از آنها از وب سرویس و استانداردهای آن در محصولات خود استفاده می کنند. **NTT Communications of Tokyo** یکی از شرکت هایی است که در حال اضافه کردن توضیحاتی به ساختار **UDDI** است. در حال حاضر شرکت ها هنوز کمی درباره وارد کردن خود در پایگاه های عمومی محتاط هستند. این چیز عجیبی نیست. شرکتها ابتدا این امکانات را فقط برای شرکای خود ایجاد می کنند. شرکتهای بزرگ نیز برای مدیریت بر سرویس های خود و اشتراک آنها بین قسمت های مختلف از این استاندارد استفاده می کنند. وقتی این استاندارد به حد بلوغ خود برسد و کاربران با آن احساس راحتی کنند استفاده از آن نیز در مکان های عمومی فراگیر خواهد بود.

این تغییر رویه برای شرکت های بزرگی که **B2B** را به روش های قدیمی اجرا کرده بودند مشکل است. بعضی نیز اشکال امنیتی بر این روش می گیرند و مایل نیستند اطلاعاتشان را بدهند. اما با گذشت زمان و کامل شدن این تکنولوژی و درک لزوم استفاده از آن شرکت ها چاره ای جز استفاده از آن ندارند.

شکل زیر نشانگر عملکرد یک وب سرویس ساده می باشد



بنابراین به زبان ساده وب سرویس درواقع یک سرویس تحت وب می باشد که با Soap یا Nusoap پیاده سازی می شود و وب سایت ها و نرم افزارهای دیگر می توانند از خدمات با سرویس های آن وب سرویس استفاده کنند و یا به وسیله آن میتوان بدون درنظر گرفتن نوع نرم افزارها آنها را با یکدیگر ترکیب کرد.

برای طراحی وب سرویس در فریم ورک گتتا به کتابخانه Nusoap نیاز دارید البته طراحی وب سرویس بدون Nusoap به وسیله Soap در پی ای قابل پیاده سازی می باشد Soap در پی ای به عنوان یک extention قابلیت فعال سازی دارد یعنی به صورت پیش فرض فعال نمی باشد و چون بسیاری از سرورها این extention را بر روی سرور فعال نمی کنند Nusoap پدید آمد تا طراحان بدون نگرانی بتوانند وب سرویس خود را تولید کنند.

نصب Nusoap در فریم ورک گتتا :

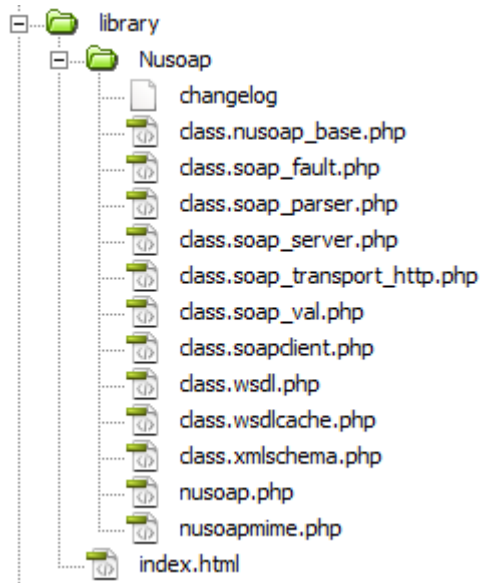
این کتابخانه بر روی سایت رسمی فریم ورک گتتا در دو نسخه GTNA Nusoap و GTNA Nusoap.rar 0.9.5 قابل دانلود است. دقت کنید که نسخه ای که بدون ذکر نسخه کتابخانه ارائه می شود حجم و قابلیت کمتری دارد ولی با تمام نسخه های فریم ورک گتتا سازگار است اما در کارهای پیشرفته و پیاده سازی وب سرویس های حرفه ای بهتر است نسخه ای که ورژن آن ذکر شده است را دانلود کنید تا بتواند از تمامی قابلیت های Nusoap استفاده کنید جهت اتصال به بانک و استفاده از وب سرویس ها از نسخه GTNA Nusoap.rar استفاده نمایید.

پس از دریافت این کتابخانه آن را در مسیر زیر از حالت فشرده خارج کنید.



فریم ورک گتنا نسخه 1.7.GFB

Application/library/



پس از انجام عملیات فوق نوبت به تست و آزمایش آن جهت عملکرد صحیح Nusoap فرا می‌رسد گتنا جهت رفاه حال طراحان وب سرویسی ساده ایجاد کرده است که به وسیله آن عملکرد صحیح Nusoap را تشخیص دهید .

آدرس وب سرویس :

<http://gtna.net/webservice/index.php?wsdl>

این وب سرویس دارای دو متد calc و hello می باشد .

متد calc: دو عدد را می‌گیرد سپس حاصل جمع آنها را بر می‌گرداند اگر مقدار یکی از پارامترها خالی باشد از سمت سرور رشته‌ی خطا ارسال می‌شود.

متد hello: یک نام از کلاینت دریافت می‌کند و پیام خوش آمد گویی را بر می‌گرداند.



کنترلر main را به صورت زیر ایجاد کنید.

```
<?php

#email:admin@gtna.net

#support:http://www.gtna.net

if( ! defined("SYSTEM_ROOT")) exit("No direct script access allowed");

class main extends GTNA_controller{

    function index(){

        $this->load->library(array('Nusoap/nusoap'));

        $client = new
        soapclient('http://www.gtna.net/webservice/index.php?wsdl',TRUE);

        // Call the SOAP method
```

```
        //$result = $client->call('hello', array('name' => 'hadi mansoori'));

        $result=$client->call('calc',array('num1'=>20,'num2'=>20));

        if($result=='err-1')

            echo "num1 or num 2 is null";

        else    print_r($result);

    }}

```

کنترلر بالا در واقع کلاینتی می‌باشد که درخواست‌ها را به سمت وب سرویس گتنا ارسال می‌کند اگر به متد calc توجه کنید دارای دو پارامتر num1 و num2 می‌باشد در صورتی که مقدار یکی از این دو پارامتر خالی باشد عملیات جمع با خطا روبرو می‌شود و از سمت سرور نام خطا ارسال بر می‌گردد که در این مثال err-1 است.



ماژول GTNA_category

با این ماژول به راحتی می توانید برای ستیت خود دسته بندی موضوعی ایجاد کنید و در هر مکانی از قالب آن را به نمایش در بیاورید.

در زیر یک نمونه کد جهت نمایش موضوعات و محتوا صفحات سایت می بینید:

```
<div id="category">
    <?php load::module('category','categories',array('test1')); ?>
</div>
<div id="pages">
    <?php
        if($data['page'])
            foreach($data['page'] as $page): ?>
                <?php echo $page->content; ?>
                <?php echo $page->date; ?>
            <?php endforeach;?>
    </div>
```

توجه : کد بالا را در تگ body قرار دهید.

تذکر: test1 نام viewی است که شما تمایل دارید محتوای آن به صفحات ارسال شود یعنی اگر تمایل دارید بعد از کلیک بر روی موضوعات صفحات آن موضوع به viewی دیگر ارسال شود می توانید نام viewی مورد نظر خود را بنویسید و سپس div صفحات یا page را در آن قرار دهید.

load::module('category','categories',array('test1'));

متغیر \$data['page'] آرایه از صفحات مربوط به موضوعی که بر روی آن کلیک شده است را در خود جای داده است.

جداول مورد نیاز جهت راه اندازی ماژول در بانک اطلاعاتی:

۱- جدول page : این جدول صفحاتی است که بنا به خواسته ی کاربر در موضوع مربوطه قرار می گیرد.

۲- جدول category : لیست تمامی موضوعات .



ساختار جدول category :

Id (PK)	cat_name	cat_tite
---------	----------	----------

Id کلید اصلی جدول category است. PK یا Primary Key به معنای کلید اصلی یا کلید اولیه می باشد بنابراین این فیلد باید کلید باشد. کلید به صفت خاصه ای می گویند که یونیک یا غیر تکرار شونده باشد.

توجه : نام فیلدهای جدول را تغییر ندهید در غیر این صورت با خطا رو به رو می شوید.

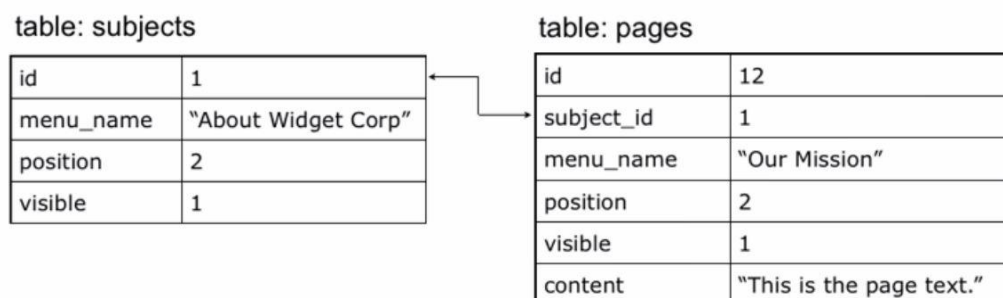
ساختار جدول page :

جدول page می تواند شامل فیلدهای دلخواه باشد اما کلید ثانویه یا کلید خارجی جدول category در این جدول باید با نام category_id ایجاد شود.

کلید خارجی یا FK :

زمانی که فیلد کلید اولیه در یک جدول با نام جدیدی در جدول دیگر قرار گیرد فیلد با نام جدید را کلید خارجی یا کلید ثانویه (FK) می نامند.

شکل زیر نمایانگر این ارتباط می باشد:

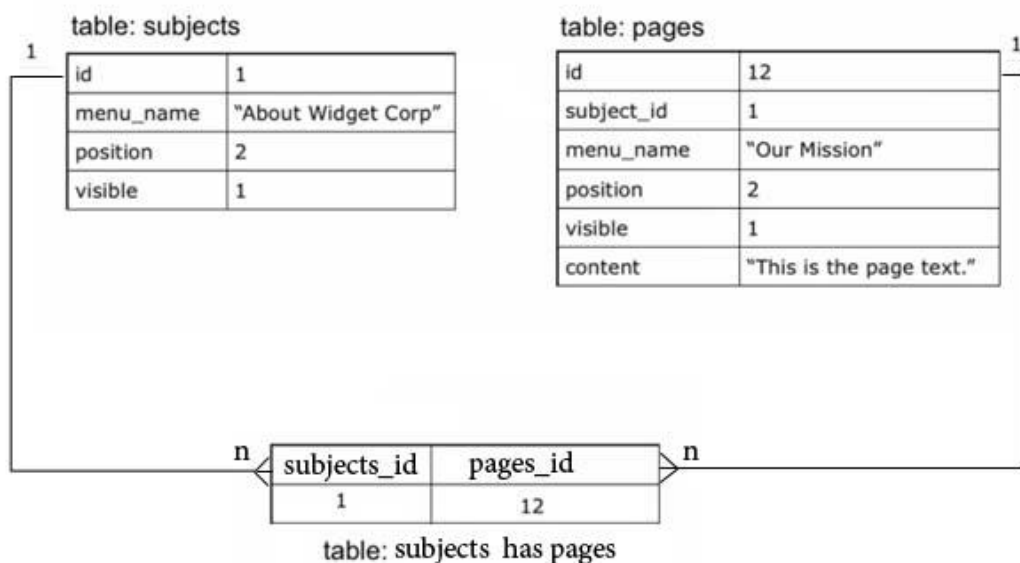


توجه : Subjects همان جدول category می باشد.

البته تصویر بالا بهینه نمی باشد. چون همواره می توانیم n موضوع و n صفحه داشته باشیم بنابراین شکل بالا بیانگر یک رابطه ی چند به چند می باشد. در مهندسی نرم افزار چنین مواقعی از یک جدول واسط استفاده می شود به شکل زیر توجه کنید:



1.7.GFB فریم ورک گتتا نسخه



طراحی رابط کاربر به وسیله‌ی CSS :

جهت زیبا سازی ظاهر مازول می توانید از CSS دلخواه استفاده کنید اما گتتا برای درک بهتر شما جهت زیبا سازی ظاهر نمونه ای را از پیش در اختیار شما قرار داده است که در زیر می توانید آن را مشاهده کنید.

```
<style>
body{
direction:rtl;
font-family:Tahoma, Geneva, sans-serif;
font-size:12px;
}

#category{
width:250px;
height:auto;
float:right;
background:#DFDFDF;
}
```



```
#category a:link,visited{
display:block;
background:#FFEDCA;
text-decoration:none;
line-height:50px;
margin-bottom:2px;
}

#category a:hover{
background:#FFE0A6;
}

#pages{
width:400px;
height:auto;
background-color:#CCC;
}
</style>
```

کدهای فوق را بعد از تگ head قرار دهید. در صورتی که از کدهای بالا درک کافی ندارید باید اطلاعات خود را در زمینه ی CSS افزایش دهید.

نصب Smarty در فریم ورک گتتا

نصب Smarty در فریم ورک گتتا

Smarty چیست؟ یک موتور قالب زنی (engine template) می باشد که کدهای ظاهر را از منطق جدا می کند. این روش در پروژه های عظیم به دلیل انعطاف در توسعه ی پروژه کاربرد دارد.

مزایای Smarty :

- ۱- سیستم Caching جهت تسریع سرعت بارگذاری صفحات.
- ۲- سیستم به صورت هوشمندانه قسمت های از صفحات که نیاز به بروز رسانی دارند را به روز می کند.
- ۳- استفاده از تگ جهت نمایش اطلاعات در قالب.
- ۴- فقط یک بار کامپایل می کند.
- ۵- سرعت تولید پروژه را کاهش می دهد.



روش نصب Smarty

تیم فنی گتنا کتابخانه‌ای جهت نصب این موتور قالب زنی برای فریم ورک گتنا تولید کرده است. اسمارتی در گتنا ملزومات متفاوتی دارد.

- ۱- موتور قالب زنی اسمارتی .
- ۲- کتابخانه‌ی GTNA_Smarty جهت فراخوانی موتور قالب زنی.
- ۳- دارای یک مثال درباره‌ی روش استفاده از موتور قالب زنی اسمارتی در فریم ورک گتنا.

دریافت اسمارتی:

پس از دریافت GTNA_Smarty.rar از سایت رسمی گتنا آن را از حالت فشرده خارج کنید. این فایل شامل موارد زیر می باشد:

- ۱- Smarty-stable: نسخه‌ی پایدار اسمارتی سازگار با فریم ورک گتنا می باشد.
- ۲- Controllers : دارای یک مثال درباره روش استفاده از این موتور قالب زنی در فریم ورک گتنا.
- ۳- GTNA_Smarty.txt : تنظیمات مربوط به موتور قالب زنی می باشد.
- ۴- GTNA_Smarty: کتابخانه‌ای طراحی شده توسط گتنا جهت بارگذاری اسمارتی می باشد.

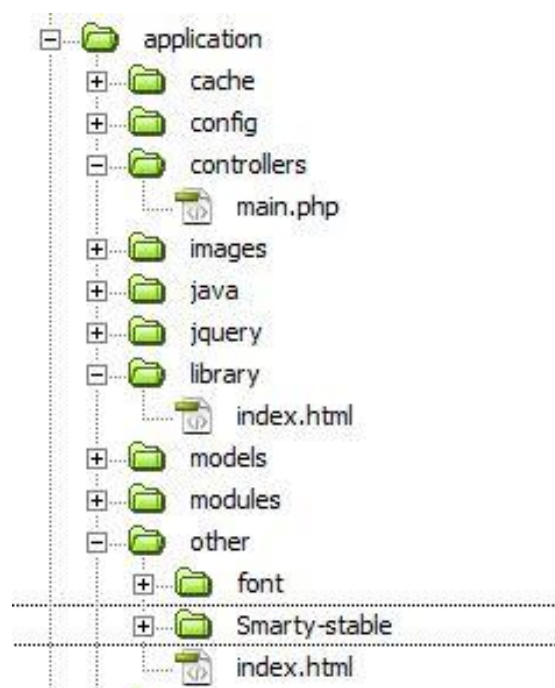
مراحل نصب اسمارتی:

- ۱- ابتدا Smarty-stable در مسیر زیر از حالت فشرده خارج کنید:

Application/other

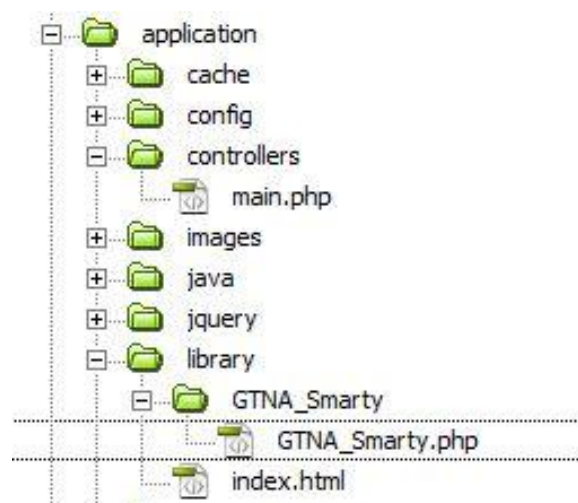


1.7. فریم ورک گنتا نسخه 1.7.GFB



۲- کتابخانه‌ی GTNA_Smarty را در مسیر زیر قرار دهید:

Application/library

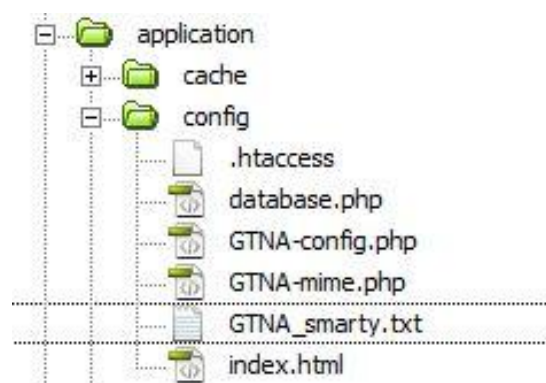


۳- فایل GTNA_Smarty.txt را در مسیر زیر قرار دهید:

Application/config



1.7. فریم ورک گنتا نسخه 1.7.GFB



۴- کنترلر main را در مسیر زیر قرار دهید:

Application/controllers



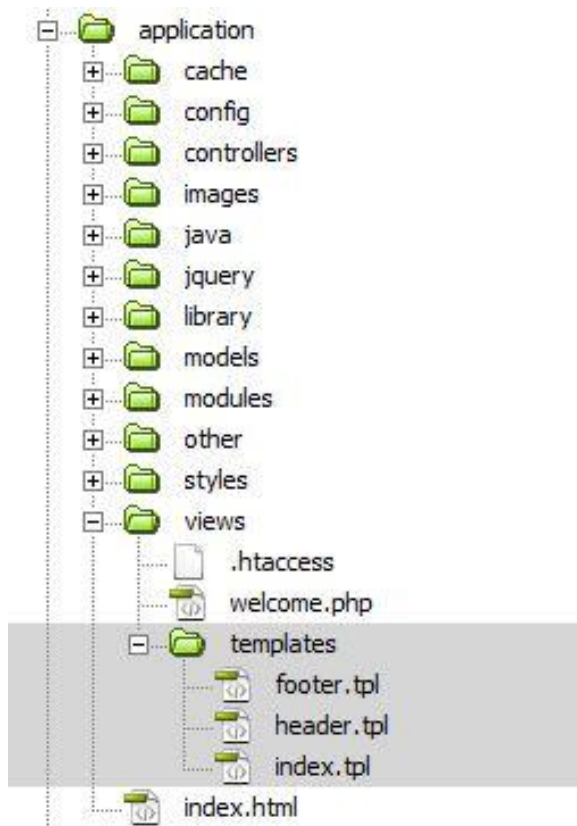
تذکر: فایل های header.tpl, index.tpl, footer.tpl را در مسیر زیر قرار دهید:

Application/views/templates/

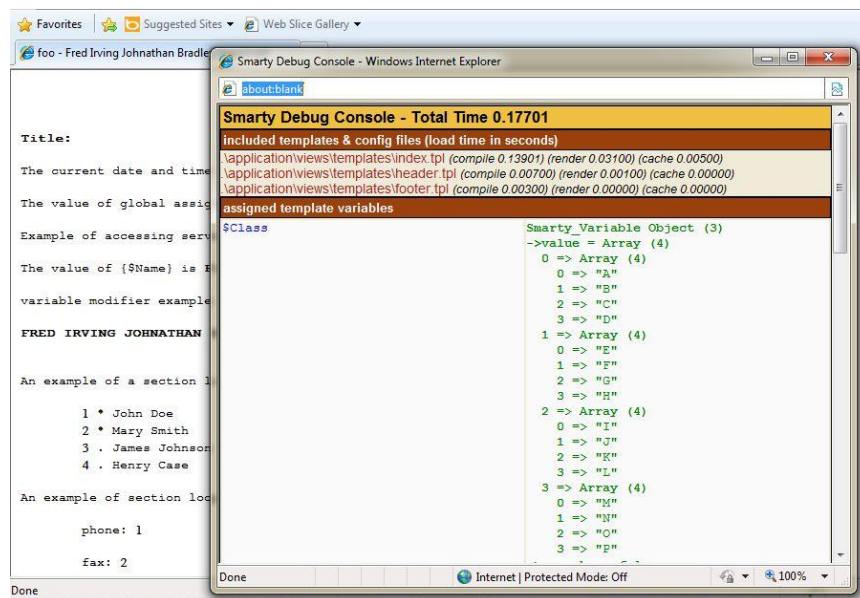
شاخه‌ی templates را در شاخه‌ی views ایجاد کنید.



1.7. فریم ورک گتتا نسخه 1.7.GFB



در صورتی که تمام مراحل را به درستی طی کرده باشید باید پس از اجرا تصویر زیر را مشاهده کنید:



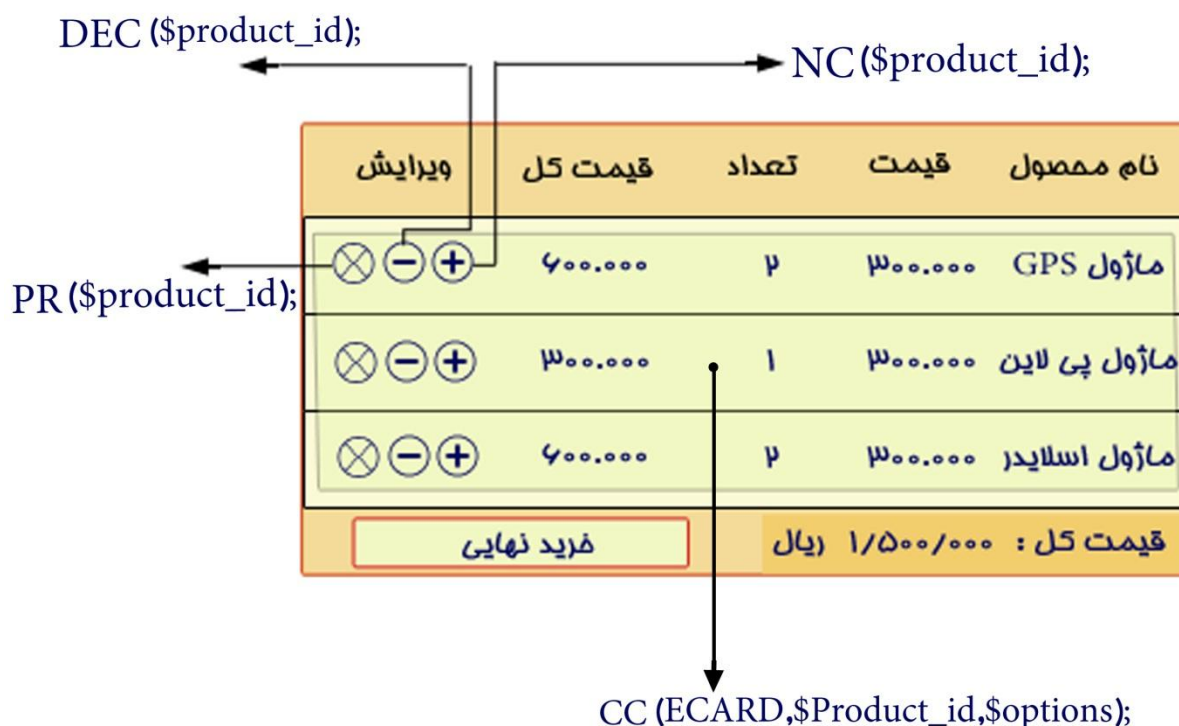


کتابخانه GTNA_Ecard

در فریم ورک گتتا نسخه هفتم طراحی سبد خرید بسیار ساده می باشد . شما فقط با ۵ متد به سادگی می توانید سبد خرید حرفه ای خود را ایجاد نمایید .

GTNA_Ecard
<code>\$this->GTNA_Ecard->CC('ECARD',\$product_id,\$options); // CARD Create</code>
<code>\$this->GTNA_Ecard->INC(\$product_id); //Increment</code>
<code>\$this->GTNA_Ecard->DEC(\$product_id); //Decrement</code>
<code>\$this->GTNA_Ecard->PR(\$product_id); //Product Remove</code>
<code>\$this->GTNA_Ecard->CR(); //CARD Remove</code>

به شکل زیر توجه کنید



۱. متد CC: در زمان فراخوانی سه آرگومان ارسال می کند

۱- آرگومان اول : در این آرگومان مقدار ECARD به عنوان یک رشته ارسال می شود در واقع این نام برای تولید سبد خرید در نشست به نام ECARD استفاده می شود توجه داشته باشد که



1.7.GFB فریم ورک گتنا نسخه

مقدار ECARD ثابت است و باید به صورت بزرگ نوشته شود و از هیچ نام دیگری نمی توانید استفاده کنید .

۲- آرگومان دوم : این آرگومان در واقع شماره محصولی در جدولی درون بانک اطلاعاتی است که خریدار بر روی آن کلیک کرده است .

۳- آرگومان سوم : این آرگومان برای ثبت مشخصات محصولات در نشست ECARD استفاده می شود فقط دقت کنید که دو فیلد **price** و **quantity** باید در آرایه **option** به صورت پیش فرض وجود داشته باشد و ساختار پایگاه داده دلخواه می باشد برای مثال می تواند درون پایگاه داده جدول محصولات را با ساختار زیر ایجاد نمایید .

Product table			
id	Product_name	price	type

نحوه عملکرد متدCC:

زمانی که خریدار بر روی یکی از محصولات کلیک می کند توسط شماره محصول می توانیم اطلاعات مورد نیاز آن محصول را واکنشی کنیم و در آرایه **options** قرار بدهیم و در نهایت آرایه را در نشست ECARD ثبت می کنیم .

پس از واکنشی اطلاعات از جدول درون پایگاه داده مقادیر مرتبط را در آرایه **options** به صورت زیر وارد می کنیم .

```
$options= array(
    'id'=>1, /*
    'product_name'=>'gtna book',
    'price'=>30000, /*
    'quantity'=>1 /*
);
```



1.7.GFB فریم ورک گتتا نسخه

اندیس های quantity و price در آرایه options اجباری و ثابت می باشند تعداد محصولات یا quantity در مرتبه اول باید ۱ باشد و بعد از کلید بر روی دگمه افزایش محصول خریدار می تواند تعداد دلخواه خود را سفارش بدهد .

توجه : قیمت کل محصولات در هیچ مکانی ذخیره نمی شود حتی در زمان تولید فاکتور برای خریدار. روش محاسبه قیمت کل برای هر محصول به صورت زیر است که توسط خود سیستم محاسبه می شود .

$Total_price = quantity * price ;$

۲. متد INC: همان طور که در شکل سبد خرید که در ابتدا آورده ایم مشاهده می کنید از این دگمه برای افزایش تعداد یک محصول خواص می باشد و بقیه موارد مثل بروزرسانی قیمت ها و تعداد توسط سیستم انجام می شود . پس از کلیک بر روی این دگمه و بدست آوردن شماره محصول عملیات افزایش محصول توسط سیستم انجام می شود .

۳. متد DEC: جهت کاهش تعداد محصول سفارش داده شده درون سبد خرید می باشد و برای این کار از شماره محصول استفاده می کند .

۴. متد PR(Product Remove) : با کلید بر روی این دگمه بر طبق شکل مشاهده می کنید که بر اساس شماره محصول آن محصول از درون نشست حذف می گردد و سپس سبد خرید توسط سیستم بروزرسانی می شود .

۵. متد CR(Card Remove) : با کلید بر روی این دکمه تمامی محصولات درون سبد خرید حذف می گردد و نیاز به هیچ آرگومانی جهت ارسال ندارد .

تذکر : برای محاسبه قیمت قابل پرداخت توسط خریدار فیلدی به نام total_price توسط سیستم به نشست ECARD اضافه می گردد کافی است که همه مقادیر این اندیس را با یکدیگر جمع کنید برای این منظور می توانید از حلقه foreach استفاده نمایید .

توجه : برای اینکه طریقه استفاده از این کتابخانه را راحت تر درک کنید گتتا کنترلی با نام ecard در سایت رسمی فریم ورک قرار داده است .



	شماره ثبت	۱۰/۱۰۴۴
	شماره شناسنامه	۸-۰۴۰۱۵-۰۳۲۱۸۸
	تاریخ ثبت	۱۳۹۲/۰۱/۲۴
فریم ورک گتنا	عنوان فارسی نرم افزار	
GTNA Framework	عنوان لاتین نرم افزار	
ندارد	عنوان فرعی نرم افزار	
۱	ویرایش نرم افزار	
ابزاری - مهندسی نرم افزار	نوع نرم افزار	
فنی، مهندسی	موضوع نرم افزار	
بزرگسالان	مخاطب نرم افزار	
این نرم افزاری به عنوان هسته کلیه برنامه های تحت وب قرار می گیرد و با کتابخانه هایی که در اختیار طراح می گذارد توسعه یک پروژه در سطح مقدماتی تا حرفه ای را بسیار آسان می کند	معرفی مختصر	
۲ مگابایت	حجم نرم افزار بر روی حامل	
اینترنت و شبکه	بستر سخت افزاری اجرای نرم افزار	
Microsoft Windows ، Linux ، Mac OS	بستر نرم افزاری اجرای نرم افزار	
به تاریخ ۱۳۹۲/۰۱/۰۷	شماره نامه تاییدیه فنی شورای عالی انفورماتیک : ۲۷۲	
مهر و امضا	مرکز توسعه فناوری اطلاعات و رسانه های دیجیتال	



بہ نام خدا

کواہی تائید فی نرم افزار

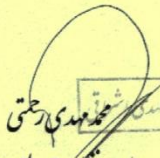
معاونت برنامه ریزی و نظارت راهبردی رئیس جمهور
بسیار عالی شورای عالی انفورماتیک کشور

نام فارسی: فریم ورک گنتا
نام لاتین: **GTNA Framework**

با استناد به مواد ۸ و ۹ قانون حمایت از حقوق پدیدآورندگان نرم افزارهای رایانه ای، مصوب ۱۳۷۹/۱۰/۴ مجلس شورای اسلامی و بخش چهارم آیین نامه اجرایی قانون مذکور تحت شماره شناسایی زیر تائید فی شد. شخصیات دارندگان حقوق مادی و معنوی (مالکین و پدیدآورندگان) در ضمیر این کواہی مندرج است.

۲ ۰ ۴ ۹ ۶ ۰

بدین است ثبت نرم افزار فوق و نشر آن تابع قوانین و مقررات مربوط می باشد.

از طرف: 
معاون نظارت راهبردی و
دبیر شورای عالی انفورماتیک کشور

این کواہی سند رسمی دولتی محسوب می شود و محرکونه. جل آن مستوجب پیگرد قانونی است.

شماره: ۲۰/۳۷-۷۰۲۵۶
تاریخ: ۲۶ اسفند ۱۳۹۱



عناوین کتاب‌های تولید شده

- ۱- سیستم عامل ویژه مقاطع کاردانی و کارشناسی (دارای مجوز وزارت ارشاد).
- ۲- فریم ورک گتنا (جلد اول) (دارای مجوز وزارت ارشاد).

عناوین کتاب‌های در دست تولید

- ۱- فریم ورک گتنا (جلد دوم).
- ۲- آموزش SEO بهینه سازی وب برای موتورهای جستجو.
- ۳- طراحی فریم ورک تحت وب بر پایه معماری سه لایه.
- ۴- تجارت الکترونیک.
- ۵- هوش مصنوعی.
- ۶- آموزش کاربردی PHP.
- ۷- فتوشاپ.

لیست نرم افزارهای جدید گتنا

- ۱- ثبت اختراع اس کیو ال فارسی گتنا بر پایه هوش مصنوعی (دارای تاییدیه شورای عالی انفورماتیک).
- ۲- فریم ورک گتنا (دارای تاییدیه شورای انفورماتیک).
- ۳- سیستم فروشگاه گتنا.
- ۴- ثبت مکانیزه متقاضیان شبکه و انشعابات شرکت گاز. (تقدیر نامه شرکت گاز)

چه تغییراتی در فریم ورک گتنا نسخه 1.7.GFB ایجاد شده است ؟

- ۱- چگونه از کتابخانه آپلودر استفاده کنیم .
- ۲- افزایش سرعت در فریم ورک گتنا .
- ۳- سیستم کش در فریم ورک گتنا.
- ۴- رفع چند باگ در کنترلر
- ۵- نصب Nusoap
- ۶- نصب Smarty
- ۷- کتابخانه GTNA_Ecard

ماژولهای طراحی شده برای فریم ورک گتنا:

- ۱- GTNA_RS : ساخت گرید ویو به همراه مدیریت رکورد ها .



۲- GTNA_Pagination: جهت صفحه بندی مطالب .

۳- GTNA_Category : برای ساخت دست بندی موضوعی و مدیریت مطالب صفحات .

۴- GTNA_Grid: برای ساخت گرید ویوی سفارشی .