Requests

Retrieving Input

You may access all input from the global Request instance.

Retrieving Data

If you have a form variable named 'address', you would want to use this code to get it:

```php
 setAddress = Request::getVar('address');
```

The DEFAULT Parameter

In the event that 'address' is not in the request or is unset, you may specify a default value as the second argument:

```php
 setAddress = Request::getVar('address', 'Address is empty');
echo setAddress;  // Address is empty
```

The SOURCE Parameter

Frequently, you will expect your variable to be found in a specific portion of the HTTP request (POST, GET, etc...). If this is the case, you should specify which portion; this will slightly increase your extension's security. If you expect 'address' to only be in POST, use this code to enforce that:

```php
 setAddress = Request::getVar('address', 'default value goes here', 'post ');
```

The VARIABLE TYPE Parameter

The fourth parameter of getVar() can be used to specify certain filters to force validation of specific value types for the variable.

```php
 setAddress = Request::getVar('address', 'default value goes here', 'post ','variable type');
```
Here is a list of types you can validate:

- INT
- INTEGER
- FLOAT
- DOUBLE
- BOOL
- BOOLEAN
- WORD
- ALNUM
- CMD
- BASE64
- STRING
- ARRAY
- PATH

**Cookies**

Cookie values may be accessed in the same manner as user input with the one change of method name from input to cookie.

```php
$value = Request::cookie('name');
```

**Other**

The Request class provides many methods for examining the HTTP request for your application.

**Request Method**

```php
$method = Request::method();

if (Request::isMethod('post'))
{
    //
}
```

**Current URL**
$current = Request::current();

**Root URL for the application**

This differs from base() in that it will always return the root URI for the application, regardless of the sub-directory the request was called from.

$base = Request::root();

**Base URL for the application**

By default, this method will return the full base path for the current request, including scheme, host, and port. To return the path only, pass true.

```
// Gets http://myhub.org/
$base = Request::base();

// Gets /
$base = Request::base(true);
```

**User IP Address**

$ip = Request::ip();