Extending the API

Plugins may extend the API simply by offering public model methods. Take for example a plugin called FooBar. The FooBar plugin may have the following model:

/plugins/foo_bar/models/foo_bar_sprocket.php

```php
<?php
class FooBarSprocket extends FooBarModel {

 public function create(array $vars) {

 }

 public function delete($sprocket_id) {

 }

 private function calculateCost($sprocket_id) {

 }
}
?>
```

FooBarSprocket offers two public methods and one private method. Both **FooBarSprocket::create()** and **FooBarSprocket::delete()** are made available via the API at the following URLs, respectively:

> ⚠ **Use CamelCase Names**
>
> We highly recommend using CamelCase when referencing models. This behavior is required for plugin models since version 4.0.

```
(Supported by all Blesta versions):
https://yourdomain.com/installpath/api/FooBar.FooBarSprocket/create.json?vars[price]=1.99
https://yourdomain.com/installpath/api/FooBar.FooBarSprocket/delete.json?sprocket_id=123

(Supported by Blesta versions pre-4.0):
https://yourdomain.com/installpath/api/foo_bar.foo_bar_sprocket/create.json?vars[price]=1.99
https://yourdomain.com/installpath/api/foo_bar.foo_bar_sprocket/delete.json?sprocket_id=123
```

> ⓘ **The API respects your authority**
>
> Only public model methods are accessible via the API, protected and private methods are not.

Each plugin may create any number of models. To avoid name conflicts with other plugins and with core Blesta models, plugin models should be named by prepending the plugin name to the model as in the example above (FooBarSprocket).