SFRA Code Quick Card

BASIC FUNCTION

```
'use strict';
//other imports here…
var server = require('server');

server.get('Start', function (req, res, next) {
//other code here…
    next();

});
module.exports = server.exports();
```

PREPEND/APPEND/REPLACE USING SUPERMODULE IN A CONTROLLER

```
var server = require('server');
server.extend(module.superModule);

server.prepend('Show', function (req, res, next) {
    //This information is inserted before the original Show Route.
    });

server.append('Show', function (req, res, next) {
    //This information is inserted after the original Show Route.
    });

server.replace('Start', function (req, res, next) {
    //This information replaces the original Start Route.
    });
```

USE MIDDLEWARE SCRIPTS & CHAINING

```
//Imports the script
var scriptName = require('*/cartridge/scripts/middleware/scriptxyz');

//Invoke the middleware step in the route
server.get('Start', scriptName.middlwareStep,
    function (req, res, next) {
    //...
});
```

CALL A SCRIPT IN A CONTROLLER (PRODUCT FACTORIES)

```
//Imports the script
var VariableName = require('*/cartridge/scripts/scriptxyz');

server.get('Show', function (req, res, next) {
    //gets the querystring parameters from req
    var params = req.querystring;
    //gets the product from ProductFactories
    var theProduct = VariableName.get(params);
    //…
});
```

SUPERMODULE CALL() METHOD TO EXTEND MODELS

```
//defines the base variable to equal module.superModule
var base = module.superModule;

//use call method to assign attribute to the object
function name(nameObject) {
    base.call(this, nameObject);
    this.newAttribute = nameObject.newAttribute;
}
store.prototype = Object.create(base.prototype);
```

TEMPLATE DECORATOR

```
<isdecorate template="common/layout/templatename">
</isdecorate>
```

LOCAL INCLUDE

```
<isinclude template="templatename"/>
```

REMOTE INCLUDE

```
<isinclude url="${URLUtils.url('ControllerName')}" />
```

CONTENT SLOT

```
<isslot id=".." description="..." context="..." />
```

FORM METADATA

```
<field formid="..." label="..." type="..." binding="..." />
```

FORM ISML

```
<div class="form-group notrequired">
    <label class="form-control-label">
        <isprint value="${pdict.parameter}" encoding="htmlcontent" />
     </label>
    <input type="checkbox" class="form-control" id="..." <isprint value="${pdict.param
</div>
```

SET CACHE IN CONTROLLER

```
var scriptName = require('*/cartridge/scripts/scriptName');

server.get('Show',scriptName.function,
    function (req, res, next) {
    res.render('templatename');
    next();
});
```