Progressive Enhancement

This functionality is Sneak Peek status. This means that the **APIs may change**. We welcome your feedback, ideas, and code, but please use caution if you use this new functionality.

On This Page

The Infusion Framework's *Progressive Enhancement* module provides support for automatic detection of browser features. The results of the checks can be used to influence the configuration of components, etc.

Basic Usage

```
fluid.enhance.check({
    check1: "my.checking.function1",
    check2: "my.checking.function2",
    ...
});
```

The function `fluid.enhance.check()` will execute the specified functions and store the results in the static environment using the associated key (e.g. `check1`). The presence of the tags in the static environment can be used in the context argument to `fluid.demands()`.

The function can also be used to register a simple boolean value, for example to distinguish a testing or development environment from a production environment, or to identify an environment for backward compatibility:

```
fluid.enhance.check({
    "fluid.uploader.fluid_1_3" : true
});
```

Framework Check Functions

The *Progressive Enhancement* module includes several functions that can check for particular features for you:

Function name	Description
`fluid.enhance.isBrowser`	Determines whether or not the environment is a web browser.
`fluid.enhance.supportsBinaryXHR`	Determines whether or not the browser supports binary XHR.
`fluid.enhance.supportsFormData`	Determines whether or not the browser supports form data.
`fluid.enhance.supportsFlash`	Determines whether or not the browser supports Flash.

These (or any subset of them) can be used as necessary, for example:

```
fluid.enhance.check({
    "fluid.browser.supportsFlash": "fluid.enhance.supportsFlash"
});
```

Clearing Results

The *Progressive Enhancement* module provides two functions that can be used to clear the results of previous checks from the static environment. These may be useful in a testing environment, for example.

Function name	Arguments	Description
`fluid.enhance.forget`	(String) `typeName`	Clears the result of the named check from the static environment.
`fluid.enhance.forgetAll`	none	Clears all of the keys added by `fluid.enhance.check()`.

Examples

```
// Define a function that determines whether or not the current browser supports video in full-screen mode.
fluid.videoPlayer.controllers.supportFullscreen = function () {
    var fullscreenFnNames = ["requestFullScreen", "mozRequestFullScreen", "webkitRequestFullScreen",
"oRequestFullScreen", "msieRequestFullScreen"];
    return fluid.find(fullscreenFnNames, function (name) {
        return !!$("<div></div>")[0][name] || undefined;
    });
};

// Register the result of the check in the static environment.
fluid.enhance.check({
    "fluid.browser.supportsFullScreen": "fluid.videoPlayer.controllers.supportFullscreen",
});

// Use the check result to enable a full-screen button only in browsers that support full-screen.
// If full-screen is not supported, the default configuration for "fullScreenButton" is an empty subcomponent.
fluid.demands("fullScreenButton", ["fluid.browser.supportsFullScreen"], {
    funcName: "fluid.toggleButton",
    container: "{controllers}.container",
    options: fullScreenButtonOptions
});
```

```
// Define a function that determines whether or not the current browser is Safari.
fluid.videoPlayer.isSafari = function () {
    var ua = navigator.userAgent.toLowerCase();
    return ((ua.indexOf("safari") > 0) && (ua.indexOf("chrome") < 0)) ? fluid.typeTag("fluid.browser.safari") :
undefined;
};

// Register the result of the check in the static environment.
fluid.enhance.check({
    "fluid.browser.safari": "fluid.videoPlayer.isSafari"
});

// Use the check result to configure a custom function that will override the default if the browser is Safari.
fluid.demands("fluid.videoPlayer.showControllers", ["fluid.browser.safari", "fluid.videoPlayer"], {
    funcName: "fluid.videoPlayer.showControllersSimple",
    args: ["{videoPlayer}"]
});
fluid.demands("fluid.videoPlayer.hideControllers", ["fluid.browser.safari", "fluid.videoPlayer"], {
    funcName: "fluid.videoPlayer.hideControllersSimple",
    args: ["{videoPlayer}"]
});
```