

OPENREFINE

EXTENSIONS

Genel olarak OpenRefine extension yapısı aşağıdaki gibidir:

```
build.xml
src/
  com/foo/bar/... *.java source files
module/
  *.html, *.vt files
  scripts/... *.js files
  styles/... *.css and *.less files
  images/... image files
  MOD-INF/
    lib/*.jar files
    classes/... java class files
    module.properties
    controller.js
```

- Module.properties adlı dosya, extension meta verilerini içerir.
- build.xml extensiondaki java kodunu çalıştırılabilir kodda derleyen 'build' işleminin bir parçasıdır.
- Controller.js adlı dosya, extension fonksiyonlarının (hook, java func.) OpenRefine'a kaydetmekten sorumludur.

Openrefine ile extension geliştirmek için openrefine'nın kaynak kodları indirilip, extensions dizini altında oluşturulmak istenen extension'nın dizini içinde derleme işlemi yapılır. Derleme işlemi için apache ant kurulumu yapılmalıdır. Sonrasında C:\Users\{yourusername}\..\OpenRefine-master\extensions\extensionname dizini altında komut satırından **ant build** komutu çalıştırılır.

Derlenen extension çalıştırabilir openrefine dizininde, webapp\extensions dizini altına kopyalanarak test edilebilir.

JAVASCRIPT & CSS DOSYALARININ YÜKLENMESİ

Openrefine da **/main/webapp/modules/core/project.vt** dosyası ana arayüz dosyasıdır. Bu dosya **velocity** template ile oluşturulmuştur ve boyutu oldukça küçüktür,. Çünkü neredeyse tüm içeriği Velocity değişkenleri **\$scriptInjection** ve **\$styleInjection** aracılığıyla dinamik olarak genişletilebilir. Kendi Javascript ve CSS

dosyalarınızın yüklenebilmesi için, /module/MOD-INF/controller.js dosyasında ClientSideResourceManager'a kaydetmeniz gerekir. Bu işlem aşağıdaki gibidir.

```
ClientSideResourceManager.addPaths(  
    "project/scripts",  
    module,  
    [  
        "scripts/foo.js",  
        "scripts/subdir/bar.js"  
    ]  
);
```

MENU EKLEMEK

Eklentinizin Javascript kodunu project.vt dosyasına dahil etmek, kodunuz kullanıcı arayüzüne fonksiyonları da kaydetmedikçe, kendi başına çok fazla bir şey yapmaz. Örneğin, bir exporterı Javascript'te uygulayabilirsiniz, ancak kullanıcı arayüzüne karşılık gelen bir menü komutu eklemezseniz, kullanıcı ihracatçıyı kullanamaz.

MAIN MENU (BUNU ÇALIŞTIRIMADIM)

Ana menü, MenuBar.appendTo, MenuBar.insertBefore ve MenuBar.insertAfter yöntemlerinden herhangi birini çağırarak eklenebilir. Her yöntem 2 parametre alır: belirli bir menü öğesini veya alt menüyü tanımlayan bir dizi ve bir adet yeni tek menü öğesi veya alt menü veya bir menü öğesi ve alt menü dizisi. Örneğin, Project > Export Filtered Rows > Templating..., menü öğesinden önce 2 menü öğesi ve bir menü ayırıcı eklemek için, bu Javascript kodunu, Javascript dosyalarınız yüklendiğinde çalıştırılacak olan yere yazın:

```
MenuBar.insertBefore(  
    ["core/project", "core/export", "core/export-templating"],  
    [  
        {  
            "label": "Menu item 1",  
            "click": function() { ... }  
        },  
        {  
            "label": "Menu item 2",  
            "click": function() { ... }  
        },  
        {} // separator  
    ]  
);
```

["Core / project", "core / export", "core / export-temprating"] menünün nereye ekleneceğini gösterir.

COLUMN HEADER MENU

Her sütun için menu ekleme işlemi main menüye göre farklılık gösterir. Çünkü, kullanıcı açılır menü düğmesini tıkladığında, belirli bir sütunun açılır menüsü anında oluşturulur.

```
DataTableColumnHeaderUI.extendMenu(function(column, columnHeaderUI, menu) {  
    MenuSystem.appendTo(menu, "", [  
        /* separator */,  
        {  
            id: "extmenu/item-1",  
            label: "**MY MENU on " + column.name,  
            click: function() {},  
            width: "350px",  
        },  
    ],  
    );  
});
```

EXTENSION BAR MENU

ExtensionBar menu ise aşağıdaki şekilde oluşturulur.

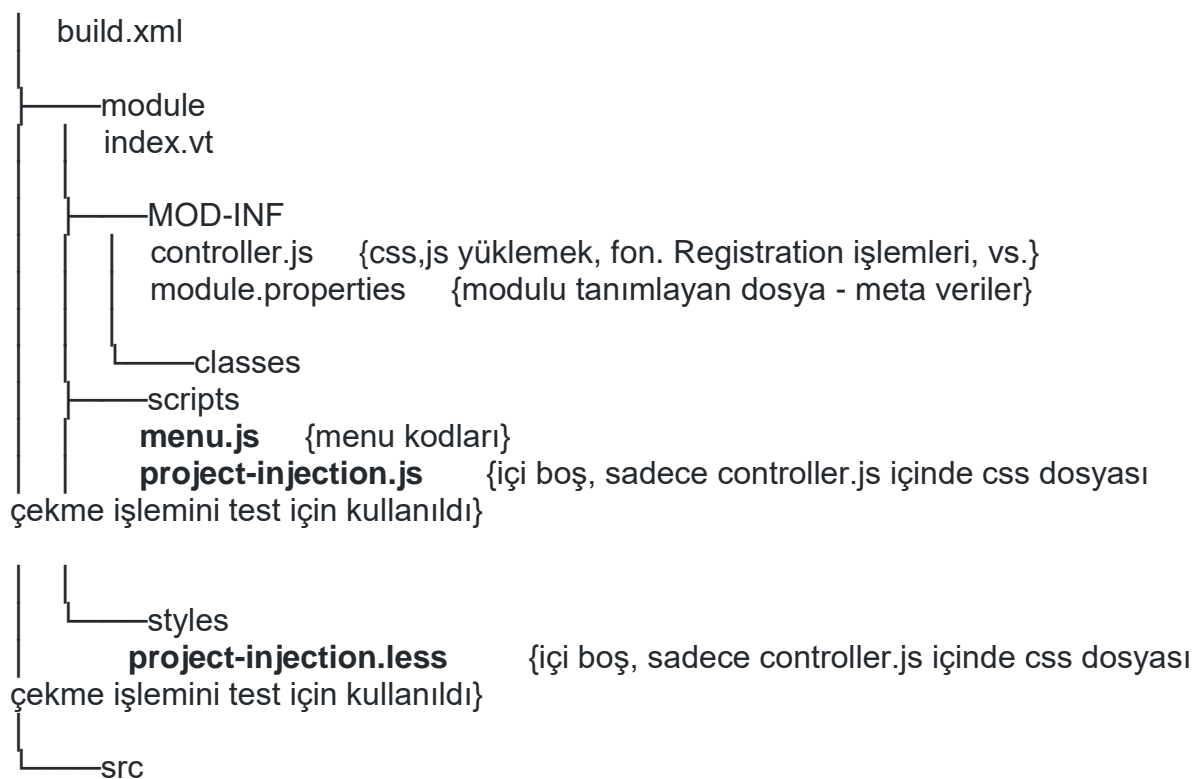
```
ExtensionBar.addExtensionMenu({  
    "id": "extbar1",  
    "label": "ExtBar Menu Items",  
    "submenu": [{  
        "id": "extbar1/item-1",  
        "label": "ExtBar Menu Item 1",  
        click: {  
            },  
    },  
    {  
        "id": "extbar1/item-2",  
        "label": "ExtBar Menu Item 2",  
        click: {}  
    },  
    ],  
});
```

OLASI ALINABİLECEK HATA

Caused by: org.mozilla.javascript.EcmaError: **ReferenceError: "ExtensionBar" is not defined.** (file:/C:/Users/HP/Desktop/openrefine-3.1/webapp/extensions/myext/module/MOD-INF/**controller.js#56**)

Controller.js içinde menu oluşturmayı deneyince yukarıdaki gibi hata alabiliriz. Bunun sebebini tam olarak bilmiyorum ama menu.js içinde menu oluşturup controller.js içinde menu.js dosyasını çekersek çalışıyor.

ÖRNEK UYGULAMANIN YAPISI



MENU.JS

```
ExtensionBar.addExtensionMenu({
    "id": "extbar1",
    "label": "ExtBar Menu Items",
    "submenu": [{
        "id": "extbar1/item-1",
        "label": "ExtBar Menu Item 1",
        click: {
```

```

        }

    },
    {},
    {
        "id": "extbar1/item-2",
        "label": "ExtBar Menu Item 2",
        click: {}
    },

]
});

DataTableColumnHeaderUI.extendMenu(function(column, columnHeaderUI, menu) {
    MenuSystem.appendTo(menu, "", [
        { /* separator */ },
        {
            id: "extmenu/item-1",
            label: "*****MY MENU on " + column.name,
            click: function() {},
            width: "350px",
        },
    ],
    });
});

```

CONTROLLER.JS

```

var html = "text/html";
var encoding = "UTF-8";
var ClientSideResourceManager =
Packages.com.google.refine.ClientSideResourceManager;

```

```

/*
 * Function invoked to initialize the extension.
 */
function init() {
    // Write to console Initializing myext extension
    Packages.java.lang.System.err.println("Initializing myext extension");
    Packages.java.lang.System.err.println(module.getMountPoint());

    // Script files to inject into /project page
    ClientSideResourceManager.addPaths(
        "project/scripts",
        module,
        [
            "scripts/project-injection.js",
            "scripts/menu.js"
        ]
    );

    // Style files to inject into /project page
    ClientSideResourceManager.addPaths(
        "project/styles",
        module,
        [
            "styles/project-injection.less"
        ]
    );

    //Packages.com.google.refine.grel.ControlFunctionRegistry.registerFunction(
    "AddTest", new Packages.io.mehmet.AddTest());
}

/*
 * Function invoked to handle each request in a custom way.
 */
function process(path, request, response) {
    // Analyze path and handle this request yourself.

    if (path == "/" || path == "") {
        var context = {};
        send(request, response, "index.vt", context);
    }
}

```

```
}  
}  
  
function send(request, response, template, context) {  
  butterfly.sendTextFromTemplate(request, response, context, template, encoding,  
html);  
}
```

KAYNAK

1. http://www.meanboyfriend.com/overdue_ideas/2017/05/writing-an-extension-to-add-new-grel-functions-to-openrefine/
2. <https://github.com/OpenRefine/OpenRefine/wiki/Write-An-Extension>
3. <https://github.com/OpenRefine/OpenRefine/wiki/Extension-Points>
4. <https://github.com/OpenRefine/OpenRefine/wiki/Sample-Extension>