

Running & Deploying your own Java Script as Extensions

To allow JavaScript to be executed as extensions during the start-up process, all you need to do is add a **.js** file to the `/ext` directory and it will be processed during startup. This is intended for once-per-session configuration of the environment.

As an example, here is an example that will allow user-defined functions in Cube and Report processing.

UserDefinedFunction.js

```
function JonFn()
{
    var self = new UserDefinedFunction("Jon's Average");
    self.m_Value = 0;
    self.m_Count = 0;
    self.clone = function()
    {
        return new JonFn();
    }
    self.update = function(data)
    {
        if (data!=null)
        {
            this.m_Value = this.m_Value + data.doubleValue();
            ++this.m_Count;
        }
    }
}
```

```

    }

    }

    self.getResult = function()

    {

    return this.m_Value / this.m_Count;

    }

    self.reset = function()

    {

    this.m_Value = 0;

    this.m_Count = 0;

    }

    return new JavaAdapter(Packages.com.elixirtech.data2["function"].Function, self);

}

importClass(Packages.com.elixirtech.data2["function"].FunctionFactory);

FunctionFactory.addUserFunction(new JonFn());

```

If you put this code in `/ext` you will find a new Function available at the bottom of the Functions list (all user-defined ones are at the bottom) called "Jon's Average". It produces exactly the same results as the built-in Average function.

The requirements for your Function implementation are that it provides the following methods:

- `clone()`
- `update(data)` // usually a Number, will be called once for each data item
- `getResult()`
- `reset()`

It can have whatever data it needs, in my example `m_Value`, `m_Count` - just use whatever names you want – there is no need to predefine.

If your function returns a different type from the input type, you will also need to supply:

`getResultType(dataType)`

which should return the `DataType` of the result, based on the `DataType` of the input. I've configured the default to be the same type (eg. `Min` on `Int` will return an `Int`, `Min` on `Date` will return a `Date` etc.)