



# System and Application Plugins

Micah Dubinko

# Writing a plugin: basics

```
import module namespace plugin =  
"http://marklogic.com/extension/plugin"  
at "/MarkLogic/plugin/plugin.xqy";
```

```
declare namespace hello = "http://marklogic.com/hello/world";
```

```
declare default function namespace  
"http://www.w3.org/2005/xpath-functions";
```

# Writing a plugin: capabilities

```
declare function hello:capabilities()  
as map:map  
{  
  let $map := map:map()  
  let $_ := map:put($map, "http://marklogic.com/appservices/  
helloworld", xdmp:function(xs:QName("hello:world")))  
  return $map  
};
```

# Writing a plugin: your stuff

```
declare function hello:world($name as xs:string?)
as xs:string
{
    if (exists($name)) then concat("Hello, ", $name) else
    "Hello!"
};
```

# Writing a plugin: registration

(: this is the body of a main module :)

```
plugin:register(hello:capabilities(), "hello-world.xqy")
```

# Calling a plugin

- Note that this code has no knowledge of available plugins...

```
let $cap-uri := "http://marklogic.com/appservices/helloworld"  
let $pid := plugin:plugins($cap-uri, "appservices")
```

(: here, handle case where \$pid has 0 or multiple :)

```
let $cap := plugin:capability($cap-uri, $pid, "appservices")  
let $result := xdmp:apply($cap, "User")
```

=>

“Hello, User”