

## index.js

```
var U = require('./user');  
var version = require('./version')();  
  
var Nick = new U.createUser('Nicholas');  
var Alex = new U.createUser('Alexander');  
  
Nick.hello(Alex);  
  
console.log(version.current);
```

## user.js


```
function User(name){  
  this.name = name;  
}  
User.prototype.hello = function(usr){  
  console.log('Hello, ' + usr.name + '!');  
};  
  
module.exports.createUser = User;
```

*экспортируется объект с методом*

## version.js

```
module.exports = function(){  
  return {'current': '1.0.0'};  
}
```

*экспортируется функция*

index.js	mysql3.js
<pre> var q = require('./mysql3').Connector;  app.get('...', function(req, res) {    var myQuery1 = "...";   q.querySelect( myQuery1,     function(err, x){       // эта коллбэк-функция       // вызывается там,       // в недрах Коннектора       // когда готов результат     }); });  app.listen(process.env.port    PORT, function(){   console.log(PORT) }); </pre>	<pre> var mysql  = require('mysql'); var conn   = mysql.createConnection( require('./bd.json') );  var Connector = (function(){   function inner(){     this.querySelect = function (myQuery, whatToDo){       // в переменную whatToDo передаём функцию       // которая сработает когда готов результат       // ...       whatToDo(..., ...);     };     this.version = function(what){       return (what) ? "latest" : "1.0.0";     };   }   return new inner; })(); exports.Connector = Connector; </pre> <p style="text-align: right;"><i>экспортируется "синглтон" с методом</i></p>
вариант: убрать Connector и экспортировать сам «синглтон»	
index2.js	mysql4.js
<pre> var q = require('./mysql4'); </pre> 	<pre> // ~/Dropbox/docs1111111/node/git/node-mysql-sessions-09-04-2016/mysql4.js // https://github.com/gossoudarev/webteach/tree/master/node-mysql-sessions-09-04-2016/mysql4.js var mysql      = require('mysql'),     conn       = mysql.createConnection( require('./bd.json') ); module.exports = (function(){   function inner(){     this.querySelect = function (myQuery, whatToDo){       //...     };   }   return new inner; })(); </pre>