

nodejs最大的优势也是大家用着最为难以理解的一点，就是它的异步功能，它几乎所有的io操作都是异步的，这也就导致很多人不理解也用不习惯。

前几天在项目中遇到这样一个问题，就是前端触发某个请求，想要查询数据库并且返回这个值，但是无论如何都返回不回来，因为没等到查询完毕，就过早的将空数据返回回来了，这个困扰了我许久，当时想到一些替代的方法，都是不治本的方法，今天打算用promise重新解决这个问题。

promise的作用是让原本异步执行的代码变成类似同步执行，就是在执行完之后，会将结果返回回来。当然，我目前也只对promise只有一个浅显的理解，在之后也会深入学习的，下面说一下这个问题是怎么解决的。

```
app.use(controller.get('/aaa', function*() {
  this.set('Cache-Control', 'no-cache');
  var data = yield service.bbb();
  this.body = data;
}));
```

我们可以使用koa框架中的yield，promise可以作为它的返回参数。

```
exports.bbb = function () {
  var promise = new Promise(function (resolve, reject) {
    var mysql = require('mysql');
    var connection = mysql.createConnection({
      host: '127.0.0.1',
      user: 'root',
      password: 'root',
      port: '3306',
      database: 'db_biology'
    });
    connection.connect();
    connection.query(
      "SELECT * FROM Sheet1",
      function selectCb(err, results) {
        if (results) {
          console.log(results);
          //resolve(results);
          resolve(results);
        }
        if (err) {
          console.log(err);
        }
        connection.end();
      }
    );
  });
  promise.then(function (value) {
    console.log(value);
    return value;
  }, function (value) {
    // failure
  });
  return promise;
};
```

只需要利用promise就可以实现我们以前直接return的结果了，这样就优雅的将异步代码变成了同步的了~