Point 1. S4 简易生成随机序列

```
[1, 2, 3, 4, 5].sort(()=> Math.random() - 0.5)
```

Point 2. 鼠标从大圆移开之后需要重置，此时需要中止之前未完成的异步请求。

这时可以调用`XMLHttpRequest.prototype.abort()`方法。

EXAMPLE：

```
var xhr = $.get('/', function(data, status) {
  // do something
})

function reset() {
  if (xhr && xhr.readyState != 4) {
    xhr.abort();
  }
}
```

*xhr 的 readyState 表示当前的状态：

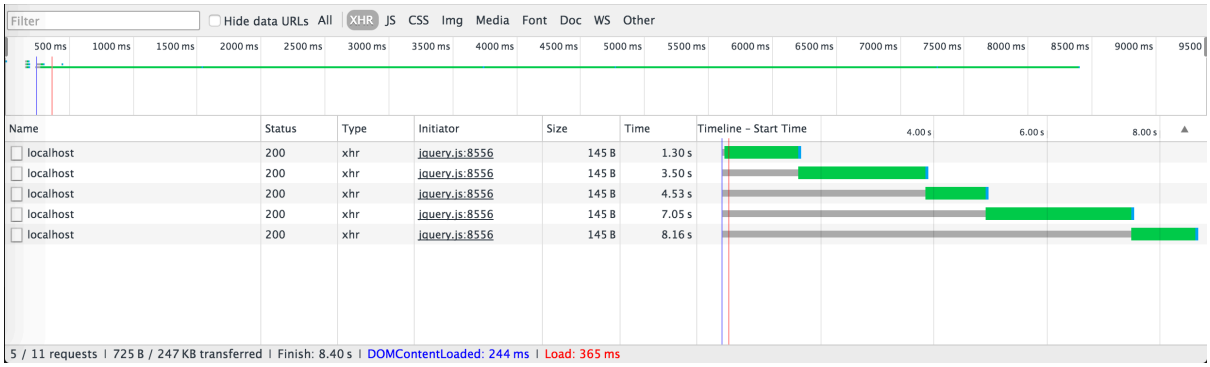
Value	State	Description
0	UNSENT	Client has been created. open() not called yet.
1	OPENED	open() has been called.
2	HEADERS_RECEIVED	send() has been called, and headers and status are available.
3	LOADING	Downloading; responseText holds partial data.
4	DONE	The operation is complete.

Point 3. Chrome 的并行 XHR 请求问题

使用 chrome 的很多同学都会在 S3 遇到不能并行 get 请求随机数的问题：

```
for (var i = 0; i < 5; i++) {
  $.get('/', function() {
    console.log(new Date());
  })
}
```

按照道理来说，的确这样写就可以并行请求了，但是执行的结果却是：



如果想看的清楚一点，可以 log 出收到回复的时间：

```
Tue Dec 15 2015 16:47:16 GMT+0800 (CST)
Tue Dec 15 2015 16:47:17 GMT+0800 (CST)
Tue Dec 15 2015 16:47:19 GMT+0800 (CST)
Tue Dec 15 2015 16:47:22 GMT+0800 (CST)
Tue Dec 15 2015 16:47:23 GMT+0800 (CST)
```

Reason：没有具体的解释，官方文档里面也找不到。之前遇到这个问题的时候曾经在 Apache 的 Document 里面找到了一段相关的文字。
http://perl.apache.org/docs/tutorials/client/browserbugs/browserbugs.html#Same_Browser_Requests_Serialization
内容大概是：

Same Browser Requests Serialization

The following feature/bug mostly affects developers.

Certain browsers will serialize requests to the same URL if accessed from different windows.

大意是说，某些特定的浏览器，比如 chrome，会阻塞 和没有响应的请求 具有相同 URL 的下一个请求。也就是说，浏览器把它给串行化(serialize)处理了。而在新版 FF 下则没有这个问题。从解决方法来看，问题的原因有可能是浏览器缓存 ajax 请求。

Sol 1. 解决方法之一是手动给请求加上时间戳：

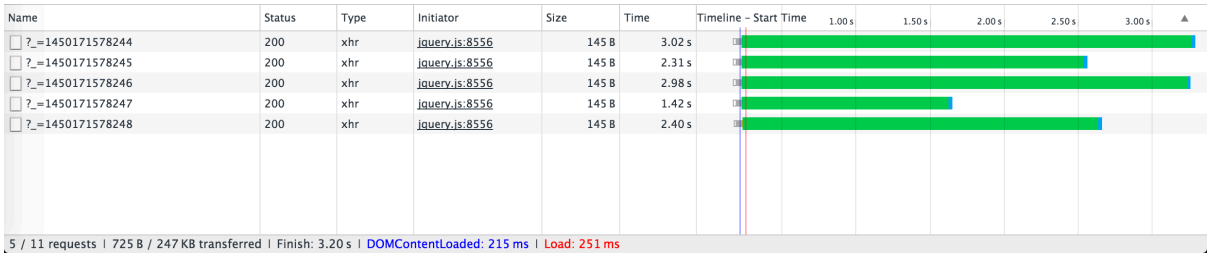
```
for (var i = 0; i < 5; i++) {
  var timestamp = new Date();
  $.get('/?date='+timestamp.getTime(), function() {
    console.log(new Date());
  })
}
```

Sol 2. 如果是使用 Jquery 的话，可以直接禁用 Ajax 缓存：

```
$.ajaxSetup({
  // 禁用缓存
  cache: false
});

for (var i = 0; i < 5; i++) {
  $.get('/', function() {})
}
```

更改后效果：



这样就能保证每一次请求的 URL 都不尽相同，从而绕开浏览器的限制。

```
Tue Dec 15 2015 17:05:37 GMT+0800 (CST)
Tue Dec 15 2015 17:05:37 GMT+0800 (CST)
Tue Dec 15 2015 17:05:38 GMT+0800 (CST)
Tue Dec 15 2015 17:05:38 GMT+0800 (CST)
Tue Dec 15 2015 17:05:38 GMT+0800 (CST)
```

Point 4. Ajax 请求的路径问题：

```
$.get("http://localhost:3000", function (result) {...
});
```

如果端口、域名或者 IP 地址改变的话，要重写所有的 ajax 请求中的 url 字段，会增加代码维护的成本。应该改为相对路径，例如：

```
"http://localhost:3000" -> "/"
"http://localhost:3000/S1" -> "/S1"
"http://localhost:3000/S1/index.html" -> "/S1/index.html"
```