class FileSystemBlobStore {

constructor(directory, prefix) {

}

has(key, invalidationKey) {

return false;

}

}

编译之后：

"use strict";

var \_createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false; descriptor.configurable = true; if ("value" in descriptor) descriptor.writable = true; Object.defineProperty(target, descriptor.key, descriptor); } } return function (Constructor, protoProps, staticProps) { if (protoProps) defineProperties(Constructor.prototype, protoProps); if (staticProps) defineProperties(Constructor, staticProps); return Constructor; }; }();

function \_classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }

var FileSystemBlobStore = function () {

function FileSystemBlobStore(directory, prefix) {

\_classCallCheck(this, FileSystemBlobStore);

}

\_createClass(FileSystemBlobStore, [{

key: "has",

value: function has(key, invalidationKey) {

return false;

}

}]);

return FileSystemBlobStore;

}();

这个可以看得出创建class就是往对象的prototype和本身添加成员

Class是不能作为函数来调用的一定要new