TEMA “UI performance”

Лектор: Нехожин Анатолий Вадимович

Дата: 06.12.2017

Netcracker

Desirable knowledge

1. JS
2. Html\css\js
3. Ajax\jquery
4. React (optional)

© Netcracker 2017

Code Performance

Example – data structure

Array initialization and benchmark find method

```
var array = [];
for (var i = 0; i < Math.pow(10, 6); ++i) {
    array.push({key: i, value: "a" + i});
}

var valueToFind = Math.ceil(Math.random()*Math.pow(10, 7));

findInArray = function () {
    var t0 = performance.now();
    var result = array.find(function (item) {
        return item.key === valueToFind;
    });
    var t1 = performance.now();

    console.log(result);
    document.getElementById("result-array").innerHTML = t1 - t0 + " ms";
};
```

Benchmark for

```
findInArray3 = function () {
    var i = 0;
    var n = array.length;
    var result;
    var t0 = performance.now();
    for (i; i < n; ++i) {
        if (array[i].key == valueToFind) {
            result = array[i];
            break;
        }
    }
    var t1 = performance.now();
};
```

Benchmark for with double equals

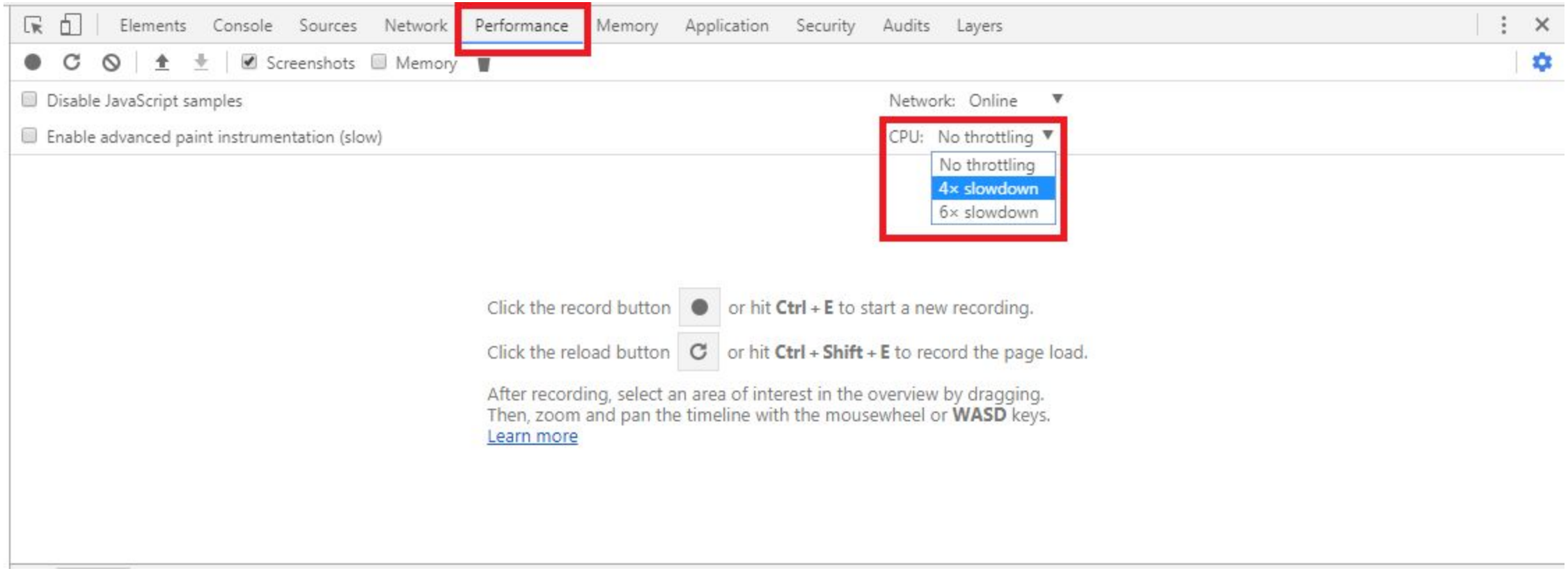
```
findInArray2 = function () {
    var i = 0;
    var n = array.length;
    var result;
    var t0 = performance.now();
    for (i; i < n; ++i) {
        if (array[i].key === valueToFind) {
            result = array[i];
            break;
        }
    }
    var t1 = performance.now();
};
```

Benchmark map

```
var map = new Map();
array.forEach(function(item) {
    map.set(item.key, item);
});

findInMap = function () {
    var t0 = performance.now();
    var result = map.get(valueToFind);
    var t1 = performance.now();
};
```


© Netcracker 2017

Throttling in devtools



© Netcracker 2017

Client Performance Techniques

Debounce

```
function attachSimpleScroll() {
    window.onscroll = function () {
        var scrolled = window.pageYOffset || document.documentElement.scrollTop;
        alert("Y scroll = " + scrolled);
    }
}

function attachDebounceScroll() {
    var timerId = null;
    window.onscroll = function () {
        clearTimeout(timerId);
        timerId = setTimeout(function () {
            var scrolled = window.pageYOffset || document.documentElement.scrollTop;
            alert("Y scroll = " + scrolled);
        }, 200);
    }
}
```


© Netcracker 2017

WebWorker

Simple JS

```
var count = Math.pow(10, 9) * 3;

function sum() {
    var t0 = performance.now();
    var result = 0;
    for (var i = 0; i < count; ++i) {
        result++;
    }
    var t1 = performance.now();
}
```

JS with WebWorker

```
self.addEventListener('message', function(e) {
    var count = e.data.count;

    var result = 0;
    for (var i = 0; i < count; ++i) {
        result++;
    }

    self.postMessage(result);
}, false);
```

```
var worker = new Worker('webworker.js');
//work only on servers, on local PC this cause security exception

function sumWebWorker() {
    worker.addEventListener('message', function (e) {
        var t1 = performance.now();
        document.getElementById("result-webWorker")
            .innerHTML = t1 - t0 + "ms (result=" + e.data + ")";
    }, false);
    var t0 = performance.now();
    worker.postMessage({count: count});
}
```


© Netcracker 2017

Defer\Async and head

Head

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script src="main.js"></script>
</head>
<body>
Head
</body>
</html>
```

Async

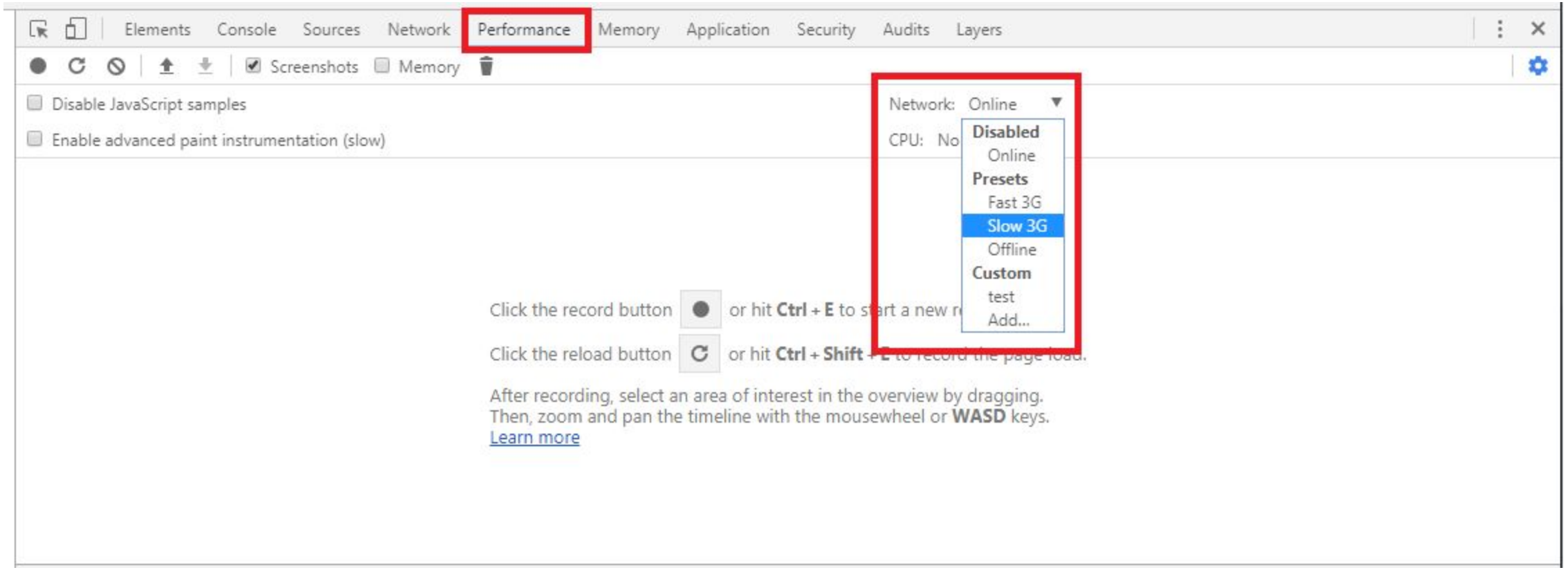
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
Async
<script src="write1.js" async></script>
<script src="write2.js" async></script>
<script src="main.js" async></script>
Async
</body>
</html>
```

Defer

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
Defer
<script src="write1.js" defer></script>
<script src="write2.js" defer></script>
<script src="main.js" defer></script>
Defer
</body>
</html>
```


© Netcracker 2017

Network deceleration in devtools

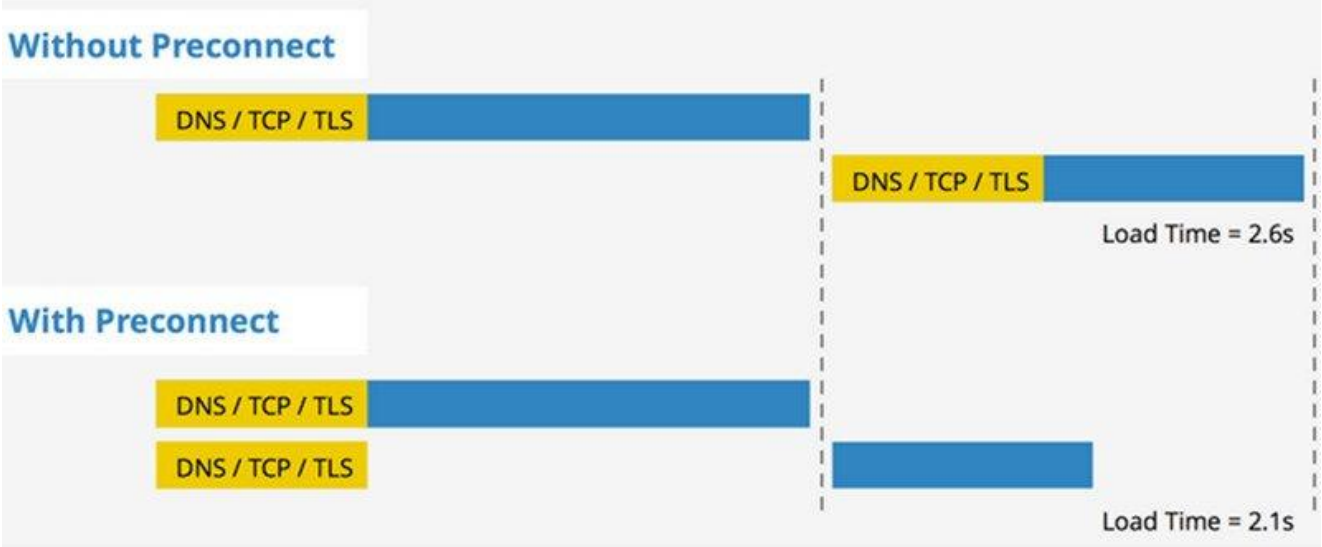


© Netcracker 2017

Resource Hints: prefetch, preconnect

- Preconnect – prepare requests in advance, e.g find dns, make tcp handshake
- Prefetch – load resources after page load finished
- Other: preload, dns-prefetch

Preconnect example



© Netcracker 2017

Minification via build tools

Techniques

- JS minification
- Styles minification
- Html minification
- JS Tree-shaking
- Image compression or compression with significant quality loose
- Bundling
- Gzip content before send from server

JS uglify example

```
//Simple calculator with add and sub methods
var calculator = {};
calculator._result = 0;
calculator.add = function (input) {
    calculator._result += input;
};
calculator.sub = function (input) {
    calculator._result = calculator._result - input;
};
calculator.showResult = function () {
    console.log("Result=" + calculator._result);
};
calculator.clear = function () {
    calculator._result = 0;
};
```

```
var calculator={};calculator._result=0,calculator.add=function(l){calculator._result+=l},
    calculator.sub=function(l){calculator._result=calculator._result-l},
    calculator.showResult=function(){console.log("Result="+calculator._result)},
    calculator.clear=function(){calculator._result=0};
```

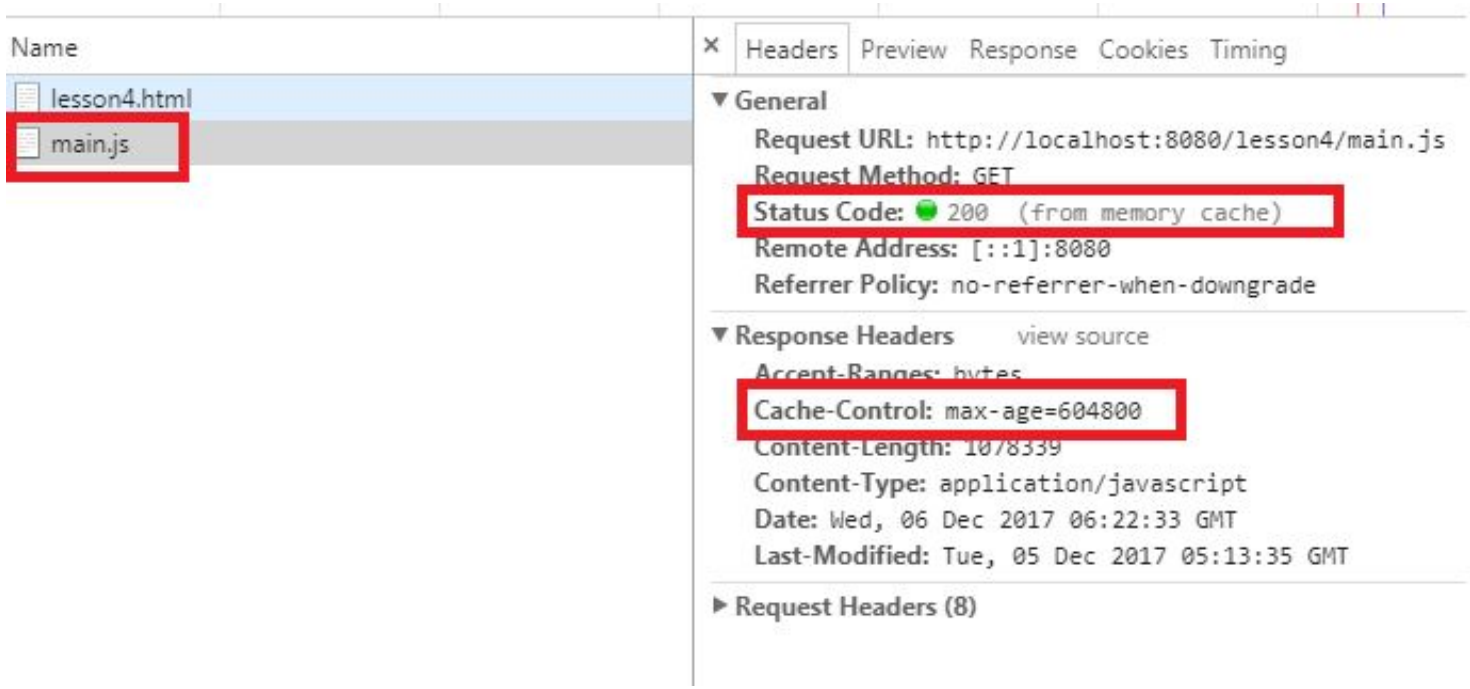

© Netcracker 2017

Minification via reducing response data

Techniques

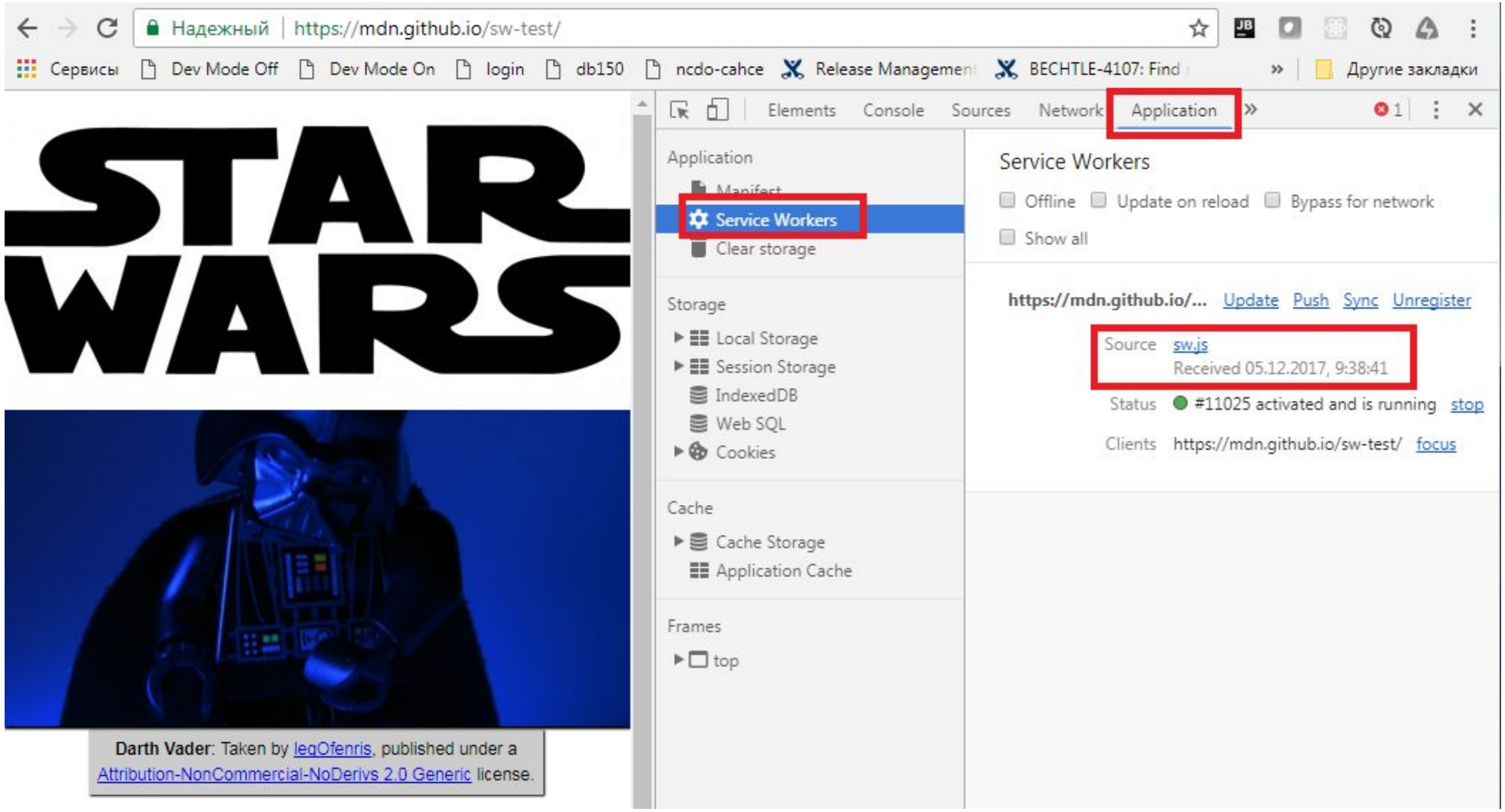
- CDN
- Code-splitting (lazy load js\css)
- Paging
- Return only needed data in response
- HTTP Cache Headers

HTTP Cache Headers Example



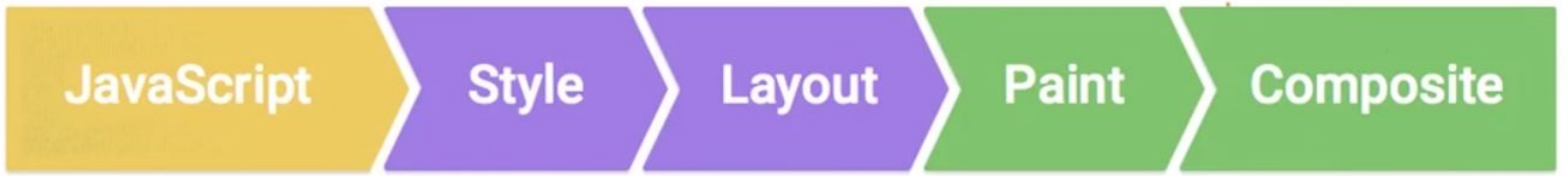
© Netcracker 2017

Service Worker for HTTPS



© Netcracker 2017

Animation



Techniques

- Reduce size and dom changing
- Use requestAnimationFrame
- Try to be in 16ms frame
- Use ccs animation
- Use dom layers

© Netcracker 2017

Animation Examples

Change size

```
function changeSize() {
    for (var i = 0; i < 10; ++i) {
        $(".size").height(30 + i * 2);
    }
}
```

CSS animation

```
function startSize() {
    $(".size").addClass("scale");
}
```

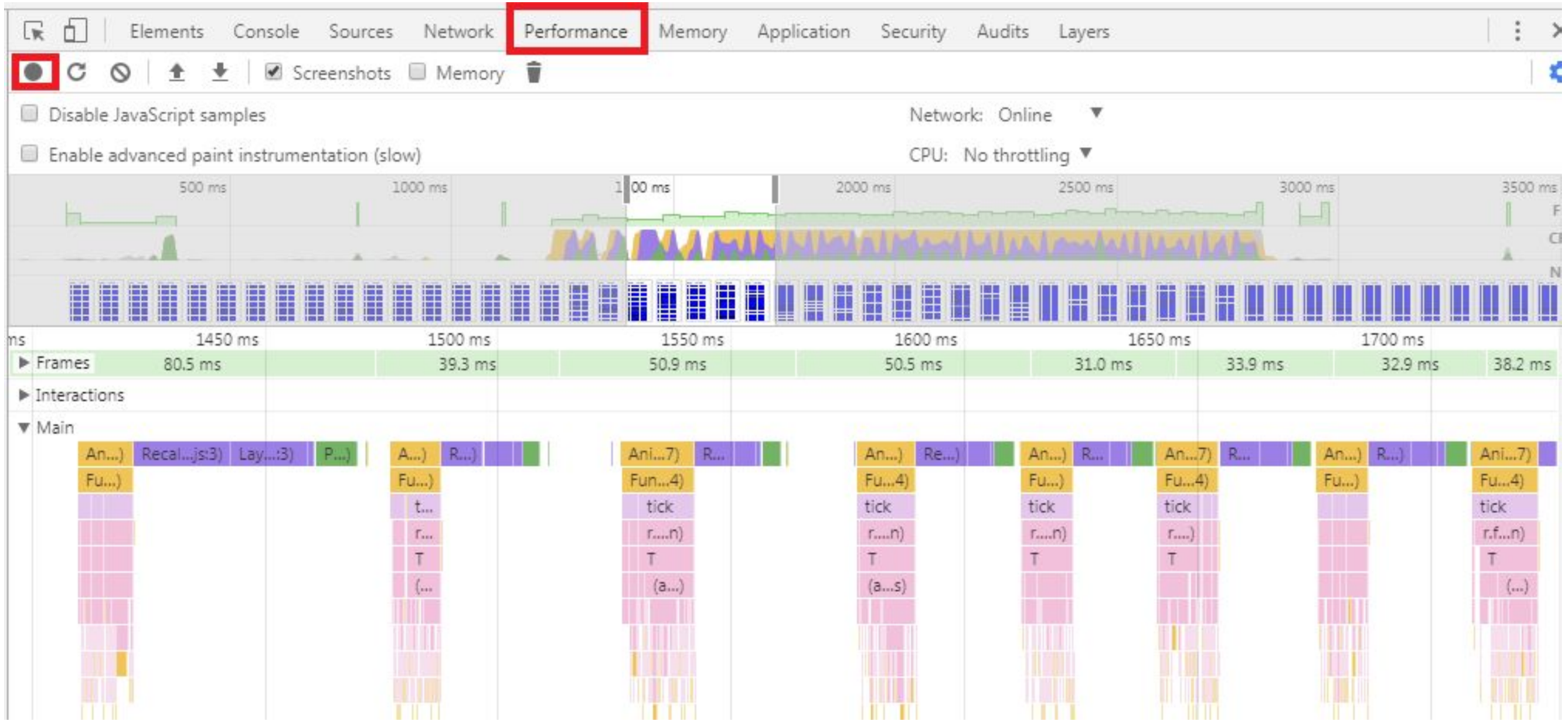
```
.scale {
    animation: run 1000ms infinite linear;
}

@keyframes run {
    0% {
        transform: scale(1);
    }
    50% {
        transform: scale(1.5);
    }
}
```

Use requestAnimationFrame

```
var frameId;
var sizeAdd = 0;

function tick() {
    ++sizeAdd;
    $(".size").height(30 + sizeAdd * 2);
    frameId = requestAnimationFrame(tick)
}


function startSize() {
    frameId = requestAnimationFrame(tick);
}

function stopSize() {
    sizeAdd = 0;
    cancelAnimationFrame(frameId);
}
```

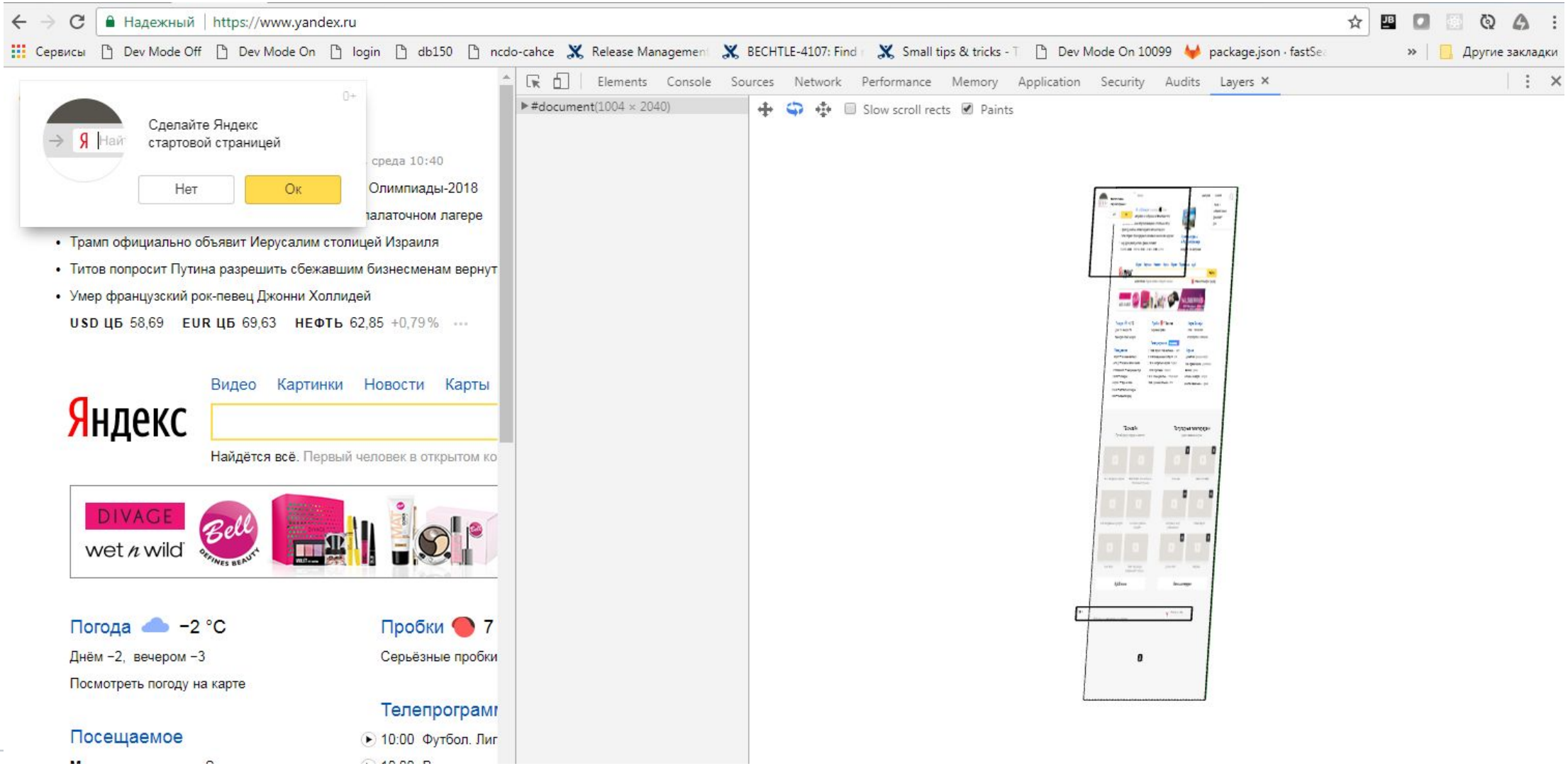

© Netcracker 2017

Performance tab



© Netcracker 2017

Dom Layers



© Netcracker 2017

Mind Tricks

Mind Tricks rules in UI

1. Что бы разница была заметна, нужна разница в 20% времени - Закон Вебера
2. Как можно быстрее перевести пользователя из пассивного ожидания в активное
3. 1.25 sec for interaction (3sec for 3G)
4. 100ms response and user response

© Netcracker 2017

Mind Tricks techniques in UI

1. Lazy loading
2. Prediction
3. SSR

© Netcracker 2017

SSR

```
const React = require("react");
const ReactDOMServer = require("react-dom/server");
const express = require('express');

const file = fs.readFileSync(__dirname + '/public/index.html', "utf8");
app.get('/dashboard/*', function (request, response) {
    let auth = JSON.parse(request.cookies["auth-token"]);
    let projectId = request.url.replace("/dashboard/", "");
    const AppServer = require("./public/react-for-server").AppServer;
    const initStore = require("./public/react-for-server").initStore;
    let projectsPromise = fetch("https://stswoon-fm-gateway.herokuapp.com/backend/project/" + auth.userId).then(res => res.json());
    let operationsPromise = fetch("https://stswoon-fm-gateway.herokuapp.com/backend/operation/" + projectId).then(res => res.json());
    let statisticPromise = fetch("https://stswoon-fm-gateway.herokuapp.com/backend/statistics/" + projectId).then(res => res.json());
    Promise.all([projectsPromise, operationsPromise, statisticPromise])
        .then(([projects, operations, statistic])  => {
            let initData = { };
            initStore(initData);
            const AppInstance2 = React.createElement(AppServer, {url: request.url}, null);
            let reactData = ReactDOMServer.renderToString(AppInstance2);
            reactData = "<div id=\"root\">" + reactData + "</div>";
            let initDataAsString = "<script>window.__initialData__ = " + JSON.stringify(initData) + "</script>";
            reactData += "\n" + initDataAsString;
            const result = file.replace("<div id=\"root\"></div>", reactData);
            response.send(result);
        })
});
```


© Netcracker 2017

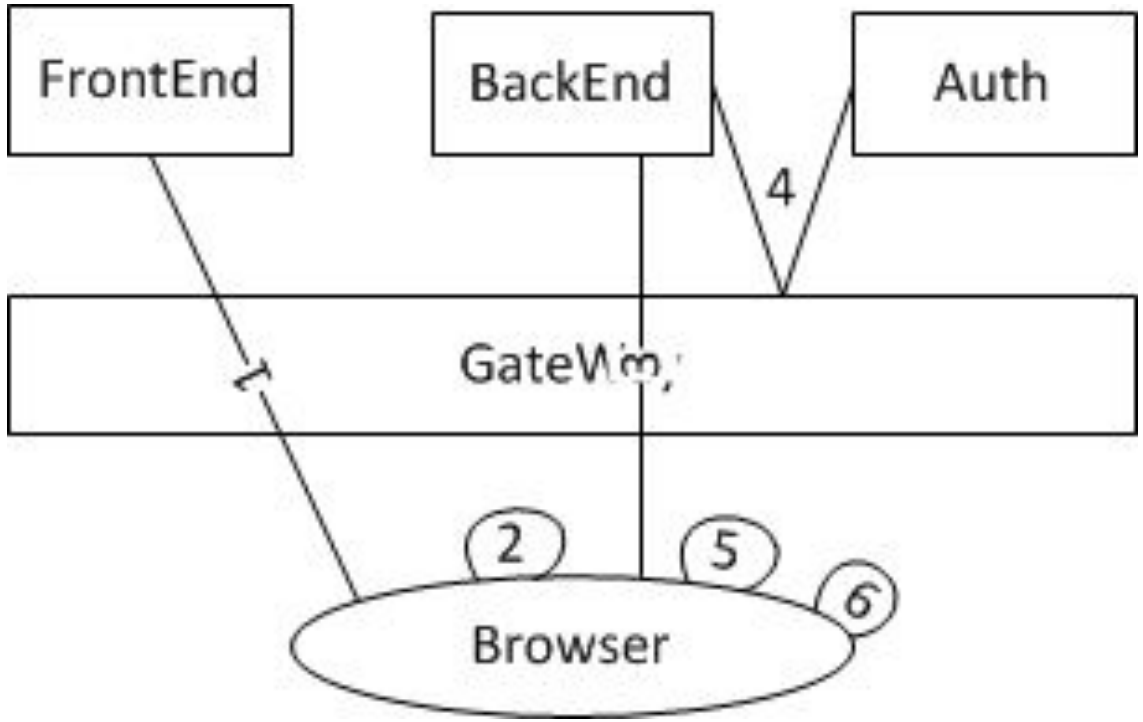
SSR

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <title>Application</title>
    <link href="/style-3ec1c52e701a446fa0fb.css" rel="stylesheet"></head>
<body>
<script>
    var envData = {};
    envData.gateway = "https://stswoon-fm-gateway.herokuapp.com";
    console.info("envData=", envData);
</script>
<div id="root"><app data-reactroot="" data-reactid="1" data-react-checksum="-1745224542"><div data-reactid="2">
    <div class="navigation" data-reactid="3"><div class="navigation_item" data-reactid="4"><div class="ProjectMenu" data-reactid="5">
        <a class="ant-dropdown-link ant-dropdown-trigger" href="#" data-reactid="6"><!-- react-text: 7 -->Projects<!-- /react-text -->
            <i class="anticon anticon-down" data-reactid="8"></i></a></div></div>
...
</app></div>
<script>window.__initialData__ = {"operations":[{"id":5,"comment":"Breakfast","operationType":"MINUS","value":150,"date":1506729600000},
    {"id":6,"comment":"Salary","operationType":"PLUS","value":1000,"date":1506729600000},{"id":7,"comment":"Clothes","operationType":"MINUS","value":250,"date":1506729600000},
    {"id":8,"comment":"Lunch","operationType":"MINUS","value":190,"date":1506729600000},{"id":9,"comment":"Additional money","operationType":"PLUS","value":30,"date":1506729600000},
    ...
</script>
<script type="text/javascript" src="/bundle-3ec1c52e701a446fa0fb.js" defer></script></body>
</html>
```

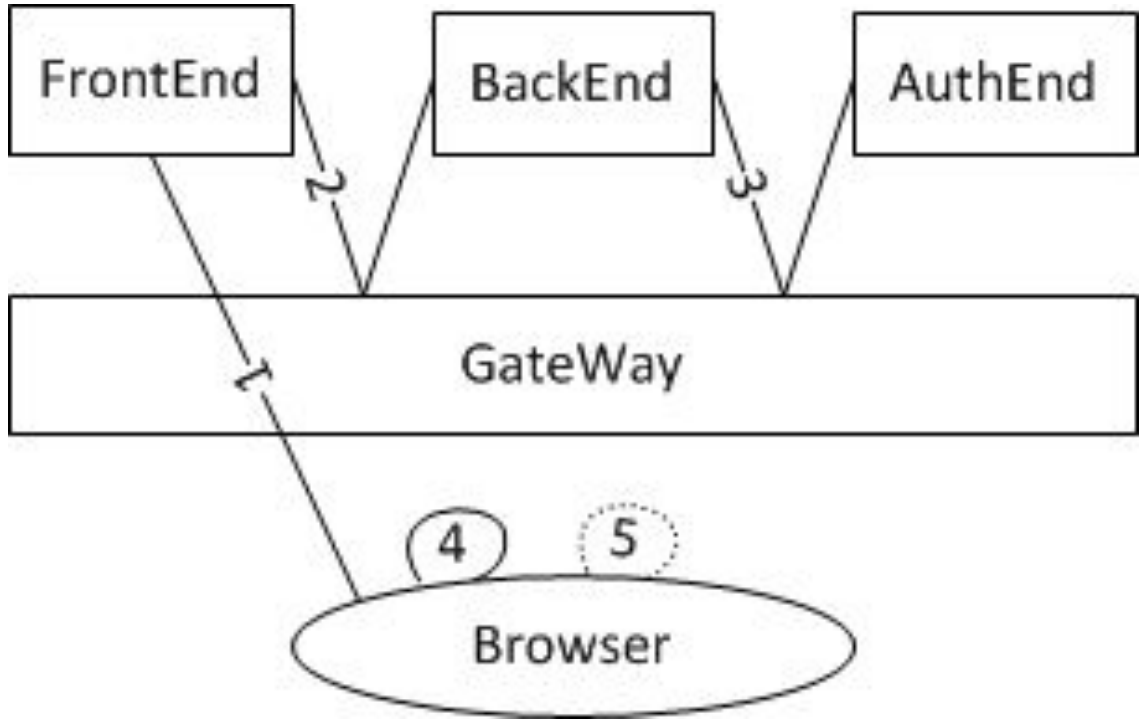

© Netcracker 2017

SSR

Client



SSR



© Netcracker 2017

Why?

Docs and Links

1. https://www.smashingmagazine.com/2016/12/front-end-performance-checklist-2017-pdf-pages/
2. https://medium.com/reloading/javascript-start-up-performance-69200f43b201
3. https://www.youtube.com/watch?v=HlKijvTa_h0
4. https://www.youtube.com/watch?v=ghcfHBEe1u4
5. https://www.youtube.com/watch?v=Tiv_9uweA7w
6. https://www.youtube.com/watch?v=lgPs_hnIA_M
7. https://www.youtube.com/watch?v=i9cOoipvST8&feature=youtu.be&list=PL8sJahqnzh8JST_ZwTcGG1FHGgKBMcpn6
8. https://www.youtube.com/watch?v=7-d3O-7aus0
9. https://www.sitepoint.com/check-css-animation-performance-with-the-browsers-dev-tools/
10. http://www.html5rocks.com/ru/tutorials/workers/basics/
11. https://www.searchengines.ru/ispolzovanie-preload-prefetch-i-preconnect.html
12. https://learn.javascript.ru/external-script
13. https://www.searchengines.ru/ispolzovanie-preload-prefetch-i-preconnect.html
14. https://developer.mozilla.org/ru/docs/Web/API/Service_Worker_API/Using_Service_Workers
15. https://css-tricks.com/the-difference-between-throttling-and-debouncing/
16. https://learn.javascript.ru/js-animation

© Netcracker 2017

Thank You