



# Building Modern Web Apps

Philip Boardman  
Sales Engineer, Sencha Inc.

# What We'll Cover

- What is a Web App?
- Loading JavaScript Libraries
- JavaScript Modules
- Build Tool
- MVVM & Data Binding
- Optimisation

# What is a Web App?

HTML + JavaScript

# HTML + JavaScript = Web App

- Make a Demo/earthquakes directory and create a file `index.html`

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Hello</title>

  <script>
    alert('Hello World');
  </script>

</head>
<body></body>
</html>
```

# HTML + JavaScript = Web App

- Make a Demo/earthquakes directory and create a file index.html

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Hello</title>

  <script>
    alert('Hello World');
  </script>

</head>
<body></body>
</html>
```



# Loading JavaScript Libraries

Bootstrapping a JavaScript environment

# Loading a JavaScript Library

- Download Ext JS from [sencha.com/products/extjs/evaluate/](http://sencha.com/products/extjs/evaluate/)
- Move Ext JS to Demo/ext
- Add <link/> to CSS file
- Add <script/> tag to load JavaScript

```
<link rel="stylesheet" type="text/css" href="../ext/build/  
classic/theme-triton/resources/theme-triton-all.css">  
<script src="../ext/build/ext-all-debug.js"></script>
```

# Using a JavaScript Library

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hello</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

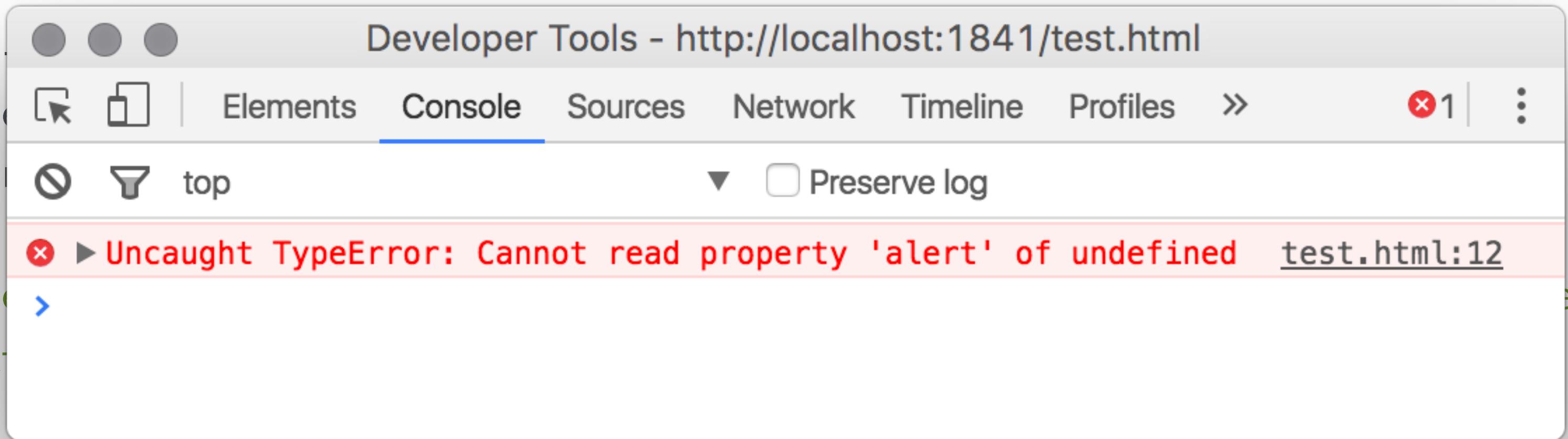
  <link rel="stylesheet" type="text/css" href="../ext/build/classic/theme-triton/resources/theme-triton-all.css">
  <script src="../ext/build/ext-all-debug.js"></script>

  <script>
    Ext.Msg.alert('Hello world', 'Welcome to Ext JS 6');
  </script>

</head>
<body></body>
</html>
```

# Using a JavaScript Library

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>Hello</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="http://extjs.com例/triton-all.css">
  <script src="../ext/ext-all.js">
    <script>
      Ext.Msg.alert('Hello world', 'Welcome to Ext JS 6');
    </script>
  </head>
  <body></body>
</html>
```



A screenshot of a browser's developer tools console window titled "Developer Tools - http://localhost:1841/test.html". The "Console" tab is selected. The error message "Uncaught TypeError: Cannot read property 'alert' of undefined" is displayed in red, along with the file path "test.html:12". The console also shows a warning icon and a "Preserve log" checkbox.

# What happened?

- Called the JavaScript library before the page has been loaded into memory
- We need to wait until document is ready before launching application
- Listen for `onready` event, handled by most JavaScript Libraries

# Waiting for onready

- Ext JS uses `Ext.application` to listen to `onready` event

```
Ext.application({  
    name: 'MyApp', // The application namespace  
    launch: function(){  
        // Run when the browser is ready  
    }  
});
```

# Ext.application

```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="UTF-8">
    <title>Hello</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" type="text/css" href="../ext/build/classic/theme-triton/resources/theme-triton-all.css">
    <script src="../ext/build/ext-all-debug.js"></script>

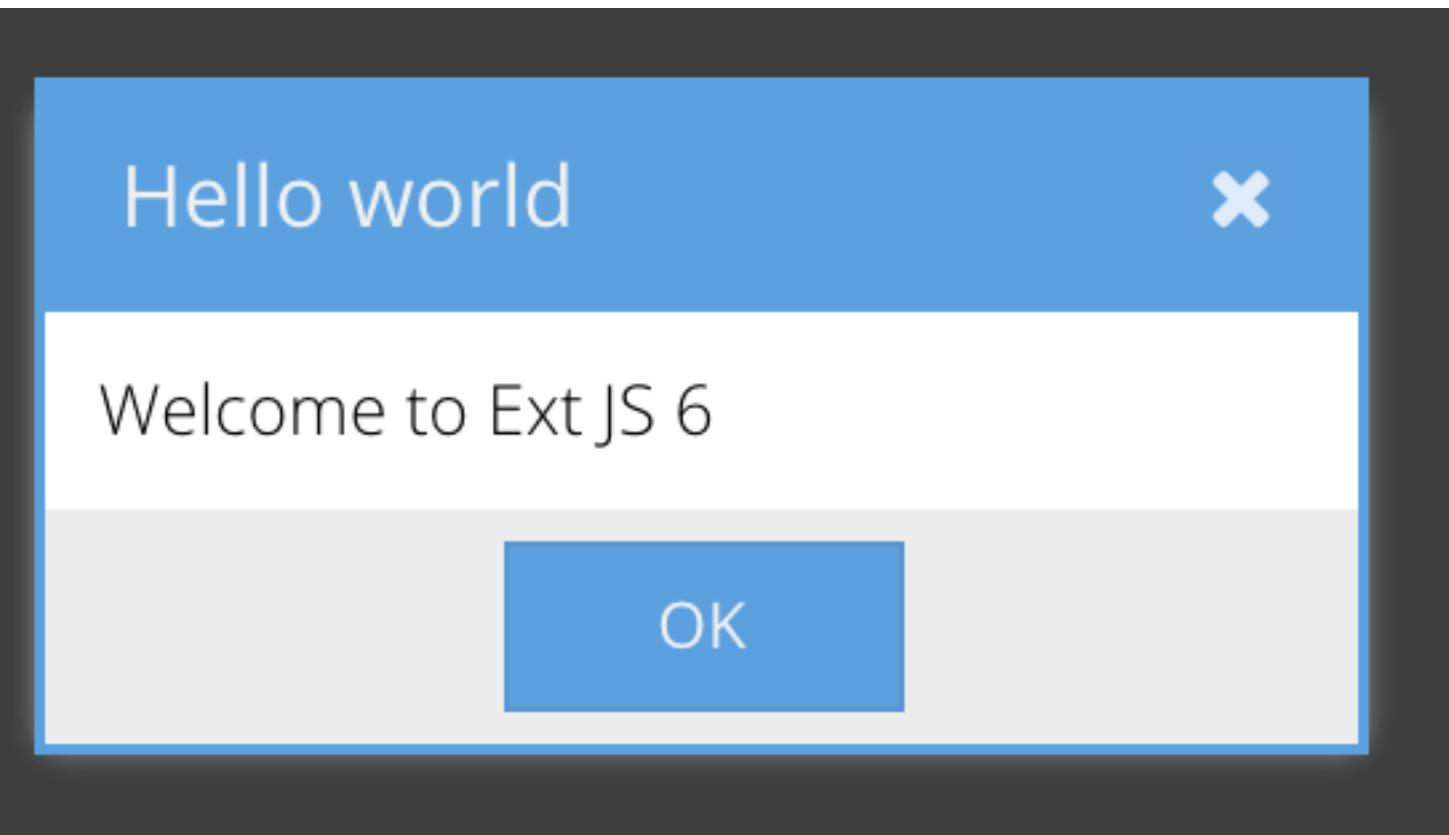
    <script>
        Ext.application({
            name: 'MyApp', // The application namespace
            launch: function(){
                Ext.Msg.alert('Hello world', 'Welcome to Ext JS 6');
            }
        });
    </script>

</head>
<body></body>
</html>
```

# Ext.application

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Hello</title>
<meta name="viewport" content="width=device-width,
<link rel="stylesheet" type="text/css" href="../ext/resources/css/ext-all.css">
<script src="../ext/build/ext-all-debug.js"></script>
<script>
Ext.application({
    name: 'MyApp', // The application namespace
    launch: function(){
        Ext.Msg.alert('Hello world', 'Welcome to Ext JS 6');
    }
});
</script>

</head>
<body></body>
</html>
```



# Increasing Complexity

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Grid</title>
<link rel="stylesheet" type="text/css" href="../ext/build/classic/theme-triton/resources/theme-triton-all.css">
<script src="../ext/build/ext-all-debug.js"></script>

<script>

Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [
        {
            xtype: 'grid',
            region: 'center',
            store: {
                model: 'Ext.data.Model',
                fields: [
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    },
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    }
                ],
                sorters: ['timestamp'],
                autoLoad: true,
                proxy: {
                    type: 'ajax',
                    url: '//apis.is/earthquake/is',
                    reader: {
                        rootProperty: 'results'
                    }
                }
            },
            columns: [
                {
                    xtype: 'datecolumn',
                    text: 'Time',
                    dataIndex: 'timestamp',
                    format: 'F j, Y \\a\\t H:i',
                    flex: 0.5
                },
                {
                    text: 'Where (V = west, A = east)',
                    dataIndex: 'humanReadableLocation',
                    flex: 1
                },
                {
                    xtype: 'numbercolumn',
                    text: 'Magnitude',
                    dataIndex: 'size',
                    width: 140,
                    align: 'right',
                    format: '0.0'
                }
            ]
        });
    Ext.application({
        name: 'Earthquakes',
        mainView: 'Earthquakes.view.main.Main'
    });
</script>
</head>
<body></body>
</html>
```

```
Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [
        {
            xtype: 'grid',
            region: 'center',
            store: {
                model: 'Ext.data.Model',
                fields: [
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    },
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    }
                ],
                sorters: ['timestamp'],
                autoLoad: true,
                proxy: {
                    type: 'ajax',
                    url: '//apis.is/earthquake/is',
                    reader: {
                        rootProperty: 'results'
                    }
                }
            },
            columns: [
                {
                    xtype: 'datecolumn',
                    text: 'Time',
                    dataIndex: 'timestamp',
                    format: 'F j, Y \\a\\t H:i',
                    flex: 0.5
                },
                {
                    text: 'Where (V = west, A = east)',
                    dataIndex: 'humanReadableLocation',
                    flex: 1
                },
                {
                    xtype: 'numbercolumn',
                    text: 'Magnitude',
                    dataIndex: 'size',
                    width: 140,
                    align: 'right',
                    format: '0.0'
                }
            ]
        });
    Ext.application({
        name: 'Earthquakes',
        mainView: 'Earthquakes.view.main.Main'
    });
});
```

45 new lines of code to add

- Panel
- Title
- Grid
- Data formatting
- Model
- Store
- Ajax Request

Increas

e to add

localhost:1841/index.html

## Earthquakes in Iceland

Time ↑	Where (V = west, A = east)	Magnitude
June 15, 2016 at 12:47	14,1 km NA af Grímsey	0.5
June 15, 2016 at 13:22	14,8 km ANA af Kistufelli	0.0
June 15, 2016 at 14:17	1,7 km VSV af Þeistareykjum	0.1
June 15, 2016 at 17:16	3,3 km S af Herðubreið	0.4
June 15, 2016 at 17:17	2,4 km NNV af Hábungu	0.1
June 15, 2016 at 21:32	31,2 km N af Siglufirði	0.5
June 15, 2016 at 21:46	27,9 km NNA af Siglufirði	0.3
June 15, 2016 at 22:18	14,6 km A af Grímsey	0.2
June 15, 2016 at 23:38	3,8 km SSA af Herðubreið	-0.2
June 16, 2016 at 03:31	5,6 km NNA af Krýsuvík	0.9
June 16, 2016 at 03:31	5,3 km NNA af Krýsuvík	0.5
June 16, 2016 at 05:12	2,1 km SSA af Hrómundartindi	1.2
June 16, 2016 at 07:34	15,1 km SSA af Grímsfjalli	0.8
June 16, 2016 at 10:08	5,5 km V af Goðabungu	0.1

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Grid</title>
<link rel="stylesheet" type="text/css" href="ext-all.css">
<script src="../ext/build/ext-all-debug.js">
<script>
Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [
        {
            xtype: 'grid',
            region: 'center',
            store: {
                model: 'Ext.data.Model',
                fields: [
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    }
                ],
                sorters: ['timestamp'],
                autoLoad: true,
                proxy: {
                    type: 'ajax',
                    url: '//apis.is/earthquake/is',
                    reader: {
                        rootProperty: 'results'
                    }
                }
            },
            columns: [
                {
                    xtype: 'datecolumn',
                    text: 'Time',
                    dataIndex: 'timestamp',
                    format: 'F j, Y \\\\'a\\\'t H:i',
                    flex: 0.5
                }, {
                    text: 'Where (V = west, A = east)',
                    dataIndex: 'humanReadableLocation',
                    flex: 1
                }, {
                    xtype: 'numbercolumn',
                    text: 'Magnitude',
                    dataIndex: 'size',
                    width: 140,
                    align: 'right',
                    format: '0.0'
                }
            ]
        });
        Ext.application({
            name: 'Earthquakes',
            mainView: 'Earthquakes.view.main.Main'
        });
    </script>
</head>
<body></body>
</html>
```

# Moving code into a separate file

- In the previous example all code is written in a single file, `index.html`
- This code can be moved to a separate file, `app.js`
- Replace inline code with a `<script>` tag

```
<script src="app.js"></script>
```

- Now our app is contained in a single external file but soon this file will become very large, what is the solution?

# JavaScript Modules

Separating code into modules allows for cleaner code

# JavaScript Modules

- Module definitions located in separate files
- Modules can be developed and tested individually
- Modules can be re-used
- Modules can be extended
- Ext JS uses classes to define modules

# Defining JavaScript Modules

- Advanced JavaScript libraries use a module definition pattern
  - *Angular* uses `angular.module`
  - *React* uses `module.exports = React.createClass`
  - *Ext JS* uses `Ext.define`
  - *Dojo Toolkit* uses Asynchronous Module Definition's (AMD) `define()`

# Separating Classes

- Our example Web App is contained within a single class
- Components which may be complex or reusable can be separated into new classes
- The Grid component can be moved from the Panel into a new class
- **xtype** links to the new class alias

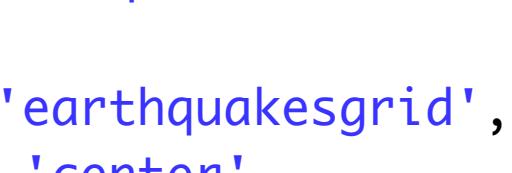
```
Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [
        {
            xtype: 'grid',
            region: 'center',
            store: {
                model: 'Ext.data.Model',
                fields: [
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    }
                ],
                sorters: ['timestamp'],
                autoLoad: true,
                proxy: {
                    type: 'ajax',
                    url: '//apis.is/earthquake/is',
                    reader: {
                        rootProperty: 'results'
                    }
                }
            },
            columns: [
                {
                    xtype: 'datecolumn',
                    text: 'Time',
                    dataIndex: 'timestamp',
                    format: 'F j, Y \\a\\t H:i',
                    flex: 0.5
                }, {
                    text: 'Where (W = west, A = east)',
                    dataIndex: 'humanReadableLocation',
                    flex: 1
                }, {
                    xtype: 'numbercolumn',
                    text: 'Magnitude',
                    dataIndex: 'size',
                    width: 140,
                    align: 'right',
                    format: '0.0'
                }
            ]
        }
});
```

# Separating Classes

- Our example Web App is contained within a single class
- Components which may be complex or reusable can be separated into new classes
- The Grid component can be moved from the Panel into a new class
- **xtype** links to the new class **alias**

```
Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [{
        xtype: 'earthquakesgrid',
        region: 'center',
        store: {
            model: 'Ext.data.Model',
            fields: [
                name: 'timestamp',
                convert: function(timestamp) {
                    return new Date(timestamp);
                }
            ],
            sorters: ['timestamp'],
            autoLoad: true,
            proxy: {
                type: 'ajax',
                url: '//apis.is/earthquake/is',
                reader: {
                    rootProperty: 'results'
                }
            }
        }
    }]
});
```

```
Ext.define('Earthquakes.view.Grid', {
    extend: 'Ext.grid.Panel',
    alias: 'widget.earthquakesgrid',
    columns: [
        xtype: 'datecolumn',
        text: 'Time',
        dataIndex: 'timestamp',
        format: 'F j, Y \\\a\\\t H:i',
        flex: 0.5
    ], {
        text: 'Where (W = west, A = east)',
        dataIndex: 'humanReadableLocation',
        flex: 1
    }, {
        xtype: 'numbercolumn',
        text: 'Magnitude',
        dataIndex: 'size',
        width: 140,
        align: 'right',
        format: '0.0'
    }
});
```



# Separating Files

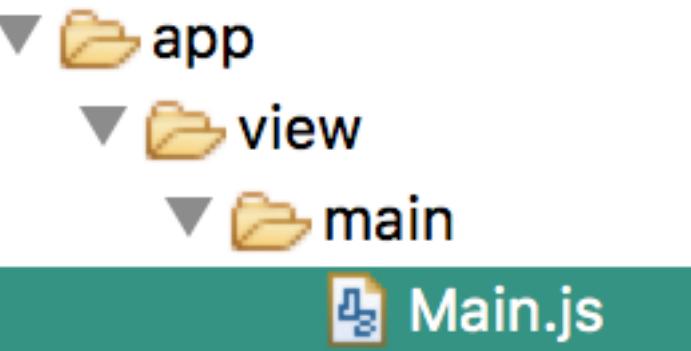
- We now have multiple class definitions within a single file.
- Separate classes into individual files
- Follow module loader naming scheme

```
Ext.define('Earthquakes.view.main.Main', {
    extend: 'Ext.panel.Panel',
    xtype: 'main',
    layout: 'border',
    title: 'Earthquakes in Iceland',
    items: [
        {
            xtype: 'earthquakesgrid',
            region: 'center',
            store: {
                model: 'Ext.data.Model',
                fields: [
                    {
                        name: 'timestamp',
                        convert: function(timestamp) {
                            return new Date(timestamp);
                        }
                    }
                ],
                sorters: ['timestamp'],
                autoLoad: true,
                proxy: {
                    type: 'ajax',
                    url: '//apis.is/earthquake/is',
                    reader: {
                        rootProperty: 'results'
                    }
                }
            }
        }
    ]
});
```

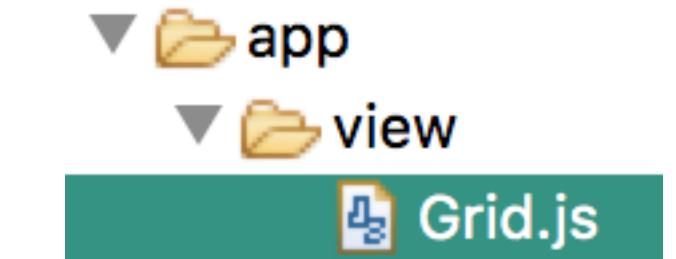
```
Ext.define('Earthquakes.view.Grid', {
    extend: 'Ext.grid.Panel',
    alias: 'widget.earthquakesgrid',
    columns: [
        {
            xtype: 'datecolumn',
            text: 'Time',
            dataIndex: 'timestamp',
            format: 'F j, Y \\\a\\\t H:i',
            flex: 0.5
        },
        {
            text: 'Where (W = west, A = east)',
            dataIndex: 'humanReadableLocation',
            flex: 1
        },
        {
            xtype: 'numbercolumn',
            text: 'Magnitude',
            dataIndex: 'size',
            width: 140,
            align: 'right',
            format: '0.0'
        }
    ]
});
```

# Separating Files

- We now have multiple class definitions within a single file.
- Separate classes into individual files
- Follow module loader naming scheme



```
Ext.define('Earthquakes.view.main.Main', {
  extend: 'Ext.panel.Panel',
  xtype: 'main',
  layout: 'border',
  title: 'Earthquakes in Iceland',
  items: [
    {
      xtype: 'earthquakesgrid',
      region: 'center',
      store: {
        model: 'Ext.data.Model',
        fields: [
          {
            name: 'timestamp',
            convert: function(timestamp) {
              return new Date(timestamp);
            }
          }
        ],
        sorters: ['timestamp'],
        autoLoad: true,
        proxy: {
          type: 'ajax',
          url: '//apis.is/earthquake/is',
          reader: {
            rootProperty: 'results'
          }
        }
      }
    }
  ]
});
```



```
Ext.define('Earthquakes.view.Grid', {
  extend: 'Ext.grid.Panel',
  alias: 'widget.earthquakesgrid',
  columns: [
    {
      xtype: 'datecolumn',
      text: 'Time',
      dataIndex: 'timestamp',
      format: 'F j, Y \\\a\\\t H:i',
      flex: 0.5
    },
    {
      text: 'Where (W = west, A = east)',
      dataIndex: 'humanReadableLocation',
      flex: 1
    },
    {
      xtype: 'numbercolumn',
      text: 'Magnitude',
      dataIndex: 'size',
      width: 140,
      align: 'right',
      format: '0.0'
    }
  ]
});
```

# Including our separate classes

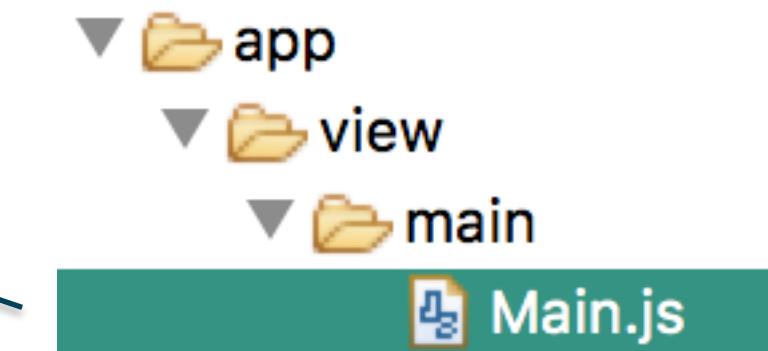
- The Web App is now spread across three files `app.js`, `Main.js` and `Grid.js`
- Each file needs to be loaded into the page explicitly
- The order of loading is important
- `app.js` requires `Main.js`
- `Main.js` requires `Grid.js`

```
<script src="app/view/Grid.js"></script>
<script src="app/view/main/Main.js"></script>
<script src="app.js"></script>
```

# Loading JavaScript Modules

- Module patterns allow code to be fetched as required by autoloader
- Module class names and file structure must match for autoloader to correctly locate modules.

```
Ext.define('Earthquakes.view.main.Main', {  
    extend: 'Ext.panel.Panel',  
    xtype: 'main',  
    layout: 'border',  
    title: 'Earthquakes in Iceland',  
    items: [  
        ...  
    ]  
});
```



# Requiring JavaScript Modules

- Modules must be required before the autoloader will fetch them
- Ext JS uses the **requires: []** attribute to specify which classes are required
- Following the naming pattern, the correct class file is located (from the path)
- If a required module has other requirements, those will also be loaded

# Requiring Classes in Ext JS

- Instead of manually loading each class file in the correct order, each class can define their own dependencies
- In `app.js` add `requires: ["Earthquakes.view.main.Main"]`
- In `Main.js` add `requires: ["Earthquakes.view.Grid"]`
- The autoloader will now fetch the appropriate class files to resolve dependencies
- Remove `Main.js` and `Grid.js` `<script>` tags from `index.html`

```
<script src="app.js"></script>
```

# That Performance Question

- Separating each class into an independent file is great for development, debugging and maintenance
- But each file is a new server request
- Each server request increases loading time
- How can we maintain the benefits of code separation while also improving performance?

# Build Tool

A build tool will development code for the production environment

# Functions of a Build Tool

- Scaffolding
- Code Generation
- Package Management
- Dependencies
- Load order
- Optimisation
- Minimisation
- Obfuscation
- Concatenation
- Caching
- JavaScript
- CSS

# Sencha Cmd

- Sencha Cmd is the build tool used for Ext JS, providing:
  - Workspace and package management
  - Scaffolding, code generation and troubleshooting
  - JS minification, optimisation and image sprinting
  - Native packaging via Cordova and PhoneGap
  - Basic web server

# Scaffolding

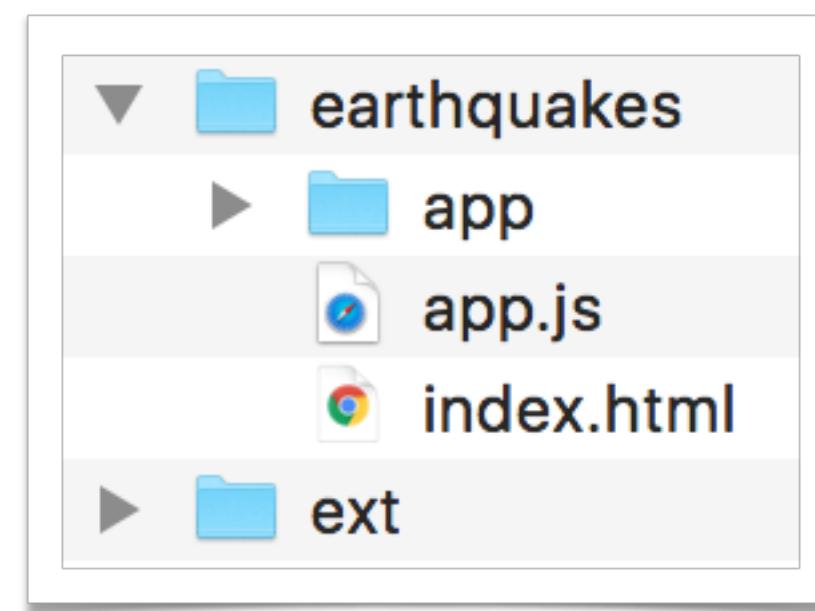
- Scaffolding allows for a quick start for simple applications
- Scaffolding generates a structure for a basic application, including:
  - File structure
  - Initial view
  - Basic controller
  - Sample data

# Scaffolding with Sencha Cmd

- Sencha Cmd offers different scaffolding options depending on the desired application
  - Using only the Classic toolkit
  - Using only the Modern toolkit
  - A Universal Web App using both Classic and Modern toolkits
- Sencha Cmd can also generate a starter application

# Viewing the Example App

- Install Sencha Cmd from [sencha.com/products/extjs/cmd-download/](http://sencha.com/products/extjs/cmd-download/)
- Place **ext** folder beside our **earthquakes** folder



- From terminal run `sencha web start` from the parent directory
- Visit `localhost:1841/earthquakes` in your browser

# Viewing

- Install Sencha Cmd

- Place example project

- From terminal

- Visit localhost:1841

The screenshot shows a web browser window titled "Earthquakes" with the URL "localhost:1841/earthquakes/". The page has a blue header bar with the title "Earthquakes in Iceland". Below the header is a table with three columns: "Time ↑", "Where (V = west, A = east)", and "Magnitude". The table lists ten earthquake events from June 15, 2016.

Time ↑	Where (V = west, A = east)	Magnitude
June 15, 2016 at 10:03	1,1 km SV af Þeistareykjum	0.9
June 15, 2016 at 10:16	14,9 km ANA af Kistufelli	0.7
June 15, 2016 at 12:47	14,1 km NA af Grímsey	0.5
June 15, 2016 at 13:22	14,8 km ANA af Kistufelli	0.0
June 15, 2016 at 14:17	1,7 km VSV af Þeistareykjum	0.1
June 15, 2016 at 17:16	3,3 km S af Herðubreið	0.4
June 15, 2016 at 17:17	2,4 km NNV af Hábungu	0.1
June 15, 2016 at 21:32	31,2 km N af Siglufirði	0.5
June 15, 2016 at 21:46	27,9 km NNA af Siglufirði	0.3
June 15, 2016 at 22:18	14,6 km A af Grímsey	0.2
June 15, 2016 at 23:38	3,8 km SSA af Herðubreið	-0.2

# Scaffolding the Example App

- Delete earthquakes/index.html
- From **ext** directory run

```
sencha generate app -classic -starter=false Earthquakes ../earthquakes
```

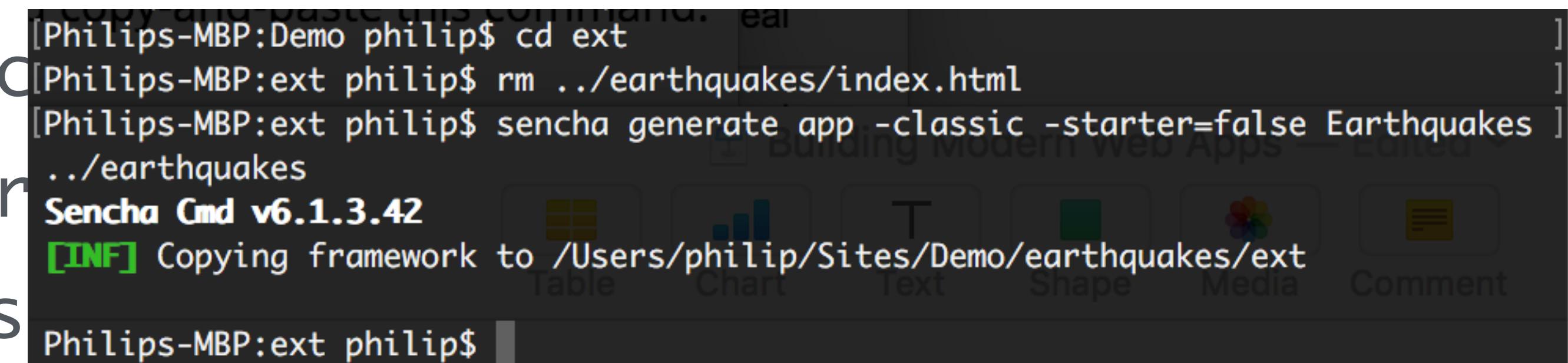
- The **-classic** parameter means only the classic toolkit is used.
- The **-starter=false** parameter means that the normal starter app will be omitted.
- **Earthquakes** is the name of the app — it's the namespace and prefix for all classes.
- **../earthquakes** is the relative path to the folder where the app is generated.
- The labs/earthquakes folder already exists, which means the existing code will be preserved.

# Scaffolding the Example App

- Delete earthquakes/index.html
- From ext directory run

```
sencha generate app -classic -starter=false Earthquakes ../earthquakes
```

- The **-classic** flag will be omitted.
- The **-starter** flag will be omitted.
- **Earthquakes** is the prefix for all classes.
- **../earthquakes** is the relative path to the folder where the app is generated.
- The labs/earthquakes folder already exists, which means the existing code will be preserved.



A screenshot of a Mac OS X desktop. In the foreground, there is a terminal window with the following command and output:

```
[Philip-MBP:Demo philip$ cd ext
[Philip-MBP:ext philip$ rm ../earthquakes/index.html
[Philip-MBP:ext philip$ sencha generate app -classic -starter=false Earthquakes
.../earthquakes
Sencha Cmd v6.1.3.42
[INF] Copying framework to /Users/philip/Sites/Demo/earthquakes/ext
Philip-MBP:ext philip$
```

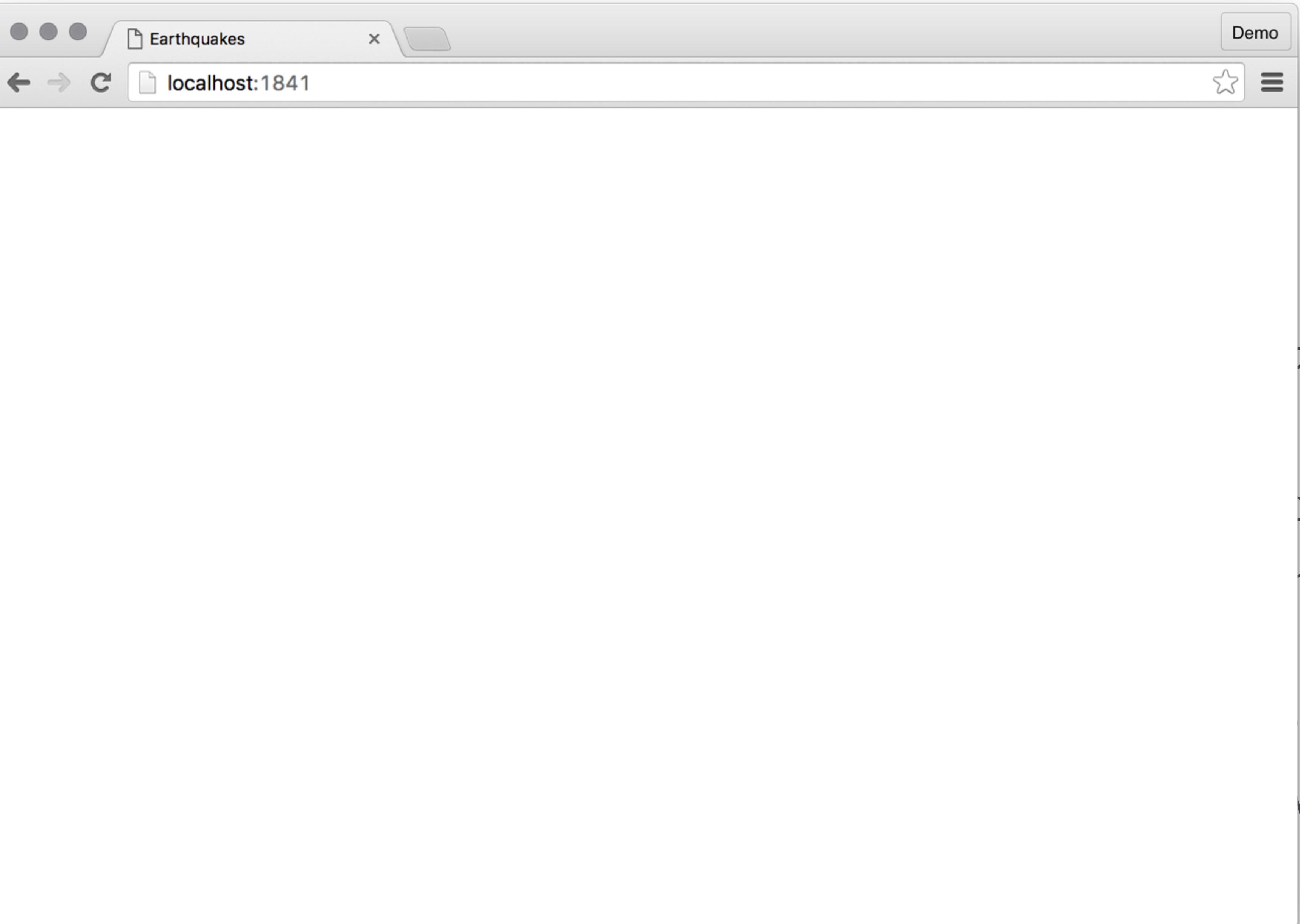
In the background, a Sencha Cmd interface window is visible, showing various icons for Table, Chart, Text, Shape, Media, and Comment.

# Scaffolding

- Delete each
- From ext

sencha gen

- The -c
  - The -s
  - Earthq
- classes
- ..../ea
- The labs/
- preserved

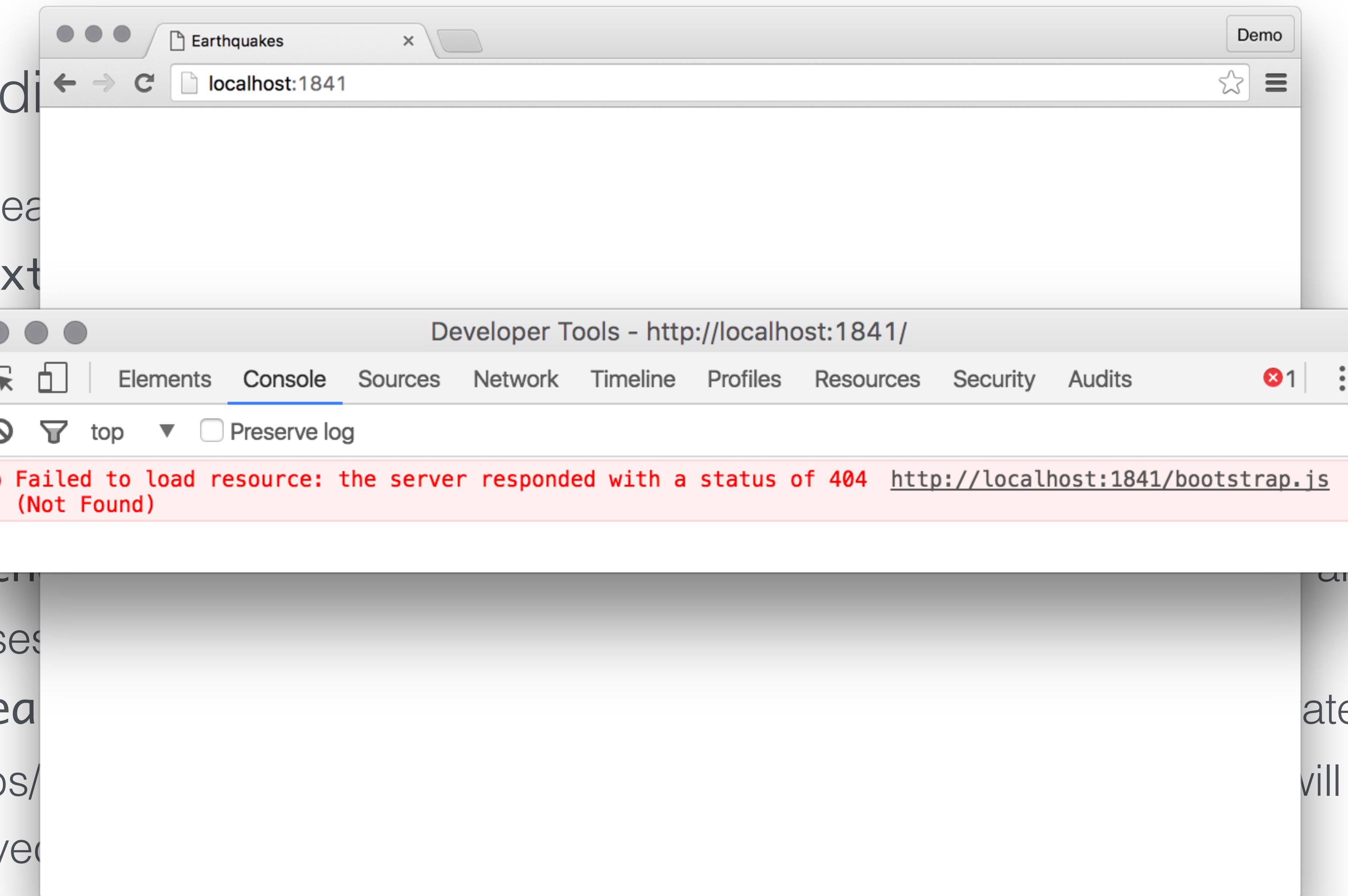


# Scaffolding

- Delete each
- From ext

sencha

- The
- The
- Ear
- classes
- ..../ea
- The labs/
- preserved



# What has Sencha CMD done?

- Our web app is broken again
- Missing file **bootstrap.js**
- Fixed by running sencha app build development
- Sencha Cmd will now read app.json and configure bootstrap.js for the development environment including
  - Theme
  - Classpath
  - External libraries
  - External CSS
  - Etc...

# What has Sencha Cmd done?

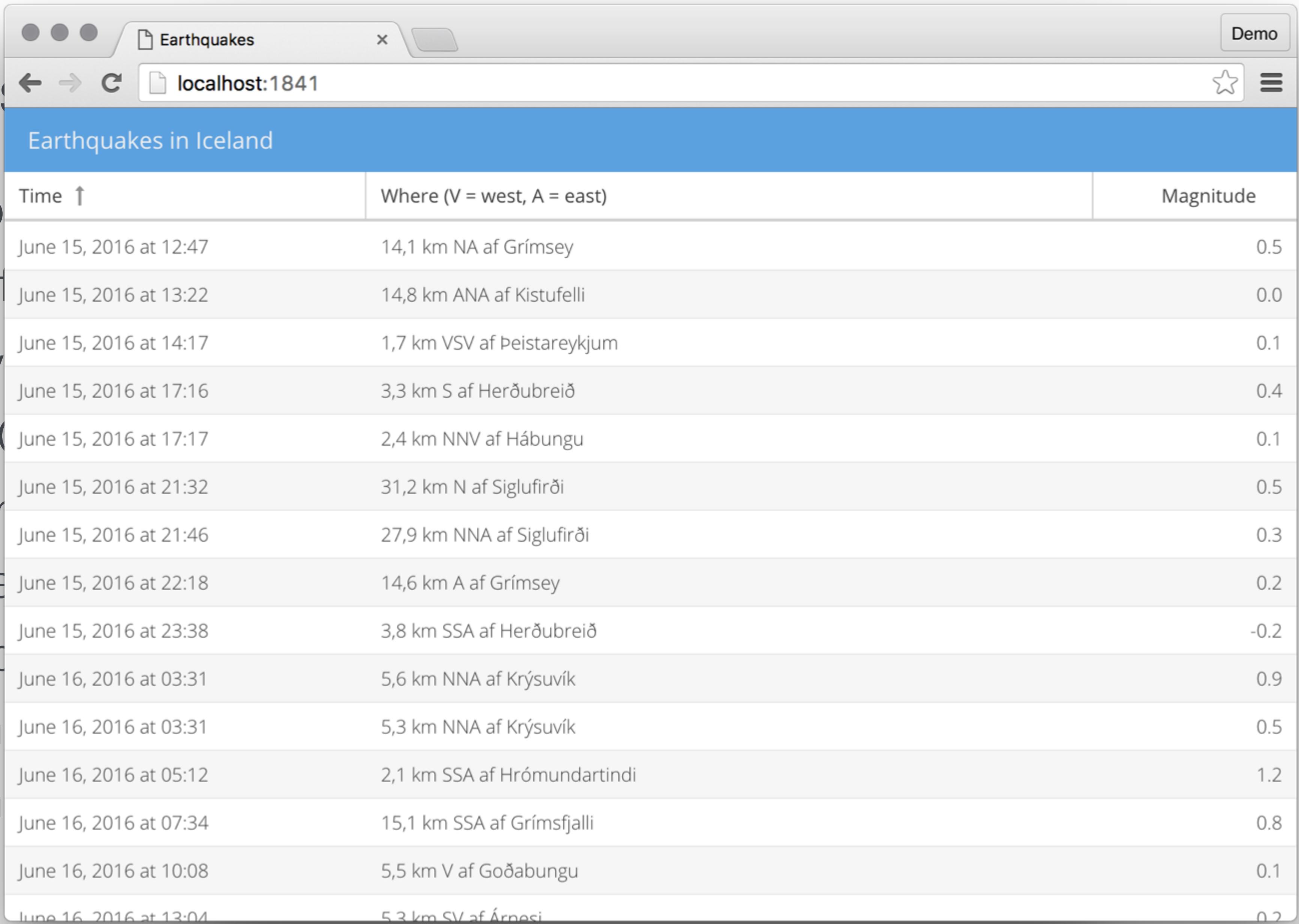
- Our web app is broken again
- Missing file `bootstrap.js`
- Fixed by running `sencha app build development`
- Sencha Cmd will now read `app.json` and configure `bootstrap.js` for the development environment including
  - Theme
  - Classpath
  - External libraries
  - External CSS
  - Etc...

```
[Philips-MBP:earthquakes philip$ sencha app build development
Sencha Cmd v6.1.3.42
[INF] Processing Build Descriptor : default
[INF] Loading app json manifest...
[INF] Appending content to /Users/philip/Sites/Demo/earthquakes/bootstrap.js
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/bootstrap.json
[INF] merging 223 input resources into /Users/philip/Sites/Demo/earthquakes/build/development/Earthquakes/resources
[INF] merged 222 resources into /Users/philip/Sites/Demo/earthquakes/build/development/Earthquakes/resources
[INF] merging 18 input resources into /Users/philip/Sites/Demo/earthquakes/build/development/Earthquakes
[INF] merged 17 resources into /Users/philip/Sites/Demo/earthquakes/build/development/Earthquakes
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.json
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.js
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/development/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] appending sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/development/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] appending sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/development/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/development/Earthquakes/sass/config.rb
[LOG] Building /Users/philip/Sites/Demo/earthquakes/build/temp/development/Earthquakes/sass/Earthquakes-all.scss
[INF] Appending content to /Users/philip/Sites/Demo/earthquakes/bootstrap.js
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/bootstrap.json

Philips-MBP:earthquakes philip$ ]
```

# What has changed?

- Our web application
- Missing features
- Fixed by Sencha GXT
- Sencha GXT development
- Themes
- Classpath
- External jars
- External dependencies
- Etc...



A screenshot of a web browser window titled "Earthquakes" at "localhost:1841". The page displays a table of 15 recent earthquakes in Iceland. The columns are "Time ↑", "Where (V = west, A = east)", and "Magnitude". The data shows events from June 15, 2016, to June 16, 2016, with magnitudes ranging from 0.0 to 1.2.

Time ↑	Where (V = west, A = east)	Magnitude
June 15, 2016 at 12:47	14,1 km NA af Grímsey	0.5
June 15, 2016 at 13:22	14,8 km ANA af Kistufelli	0.0
June 15, 2016 at 14:17	1,7 km VSV af Þeistareykjum	0.1
June 15, 2016 at 17:16	3,3 km S af Herðubreið	0.4
June 15, 2016 at 17:17	2,4 km NNV af Hábungu	0.1
June 15, 2016 at 21:32	31,2 km N af Siglufirði	0.5
June 15, 2016 at 21:46	27,9 km NNA af Siglufirði	0.3
June 15, 2016 at 22:18	14,6 km A af Grímsey	0.2
June 15, 2016 at 23:38	3,8 km SSA af Herðubreið	-0.2
June 16, 2016 at 03:31	5,6 km NNA af Krýsuvík	0.9
June 16, 2016 at 03:31	5,3 km NNA af Krýsuvík	0.5
June 16, 2016 at 05:12	2,1 km SSA af Hrómundartindi	1.2
June 16, 2016 at 07:34	15,1 km SSA af Grímsfjalli	0.8
June 16, 2016 at 10:08	5,5 km V af Goðabungu	0.1
June 16, 2016 at 13:04	5,3 km SV af Árnæsi	0.2

# What else has Sencha Cmd done?

- Copied `ext` within our working directory
- Created `build`, `packages` and `sass` (theming) directories
- Added `index.html` and configuration files
  - `app.json`
  - `bootstrap.js`
  - `bootstrap.json`
  - `build.xml`
  - `workspace.json`
- After configuring the environment and external resources, 95% of time is still spent working on our code in our `app` directory

# The new Structure

- **index.html** is a fully-formed HTML document include appropriate `<meta/>` tags to support mobile devices
- Remove references to stylesheet and JavaScript files
- Use **bootstrap.js** *microloader* to intelligently fetch appropriate resources
- **bootstrap.js** was generated by *Sencha Cmd* by reading **app.json** and **bootstrap.json**
- The *microloader* will also use application caching to speed up loading
- **build.xml** is an ANT build file to define build targets
- **workspace.json** specifies the location of code packages and the build

# MVVM & Data Binding

Model, View and View Model

# MVVM - Model, View and ViewModel

- By introducing the ViewModel as an intermediary step we can share common logic and data between multiple views
- The MVVM pattern can assist when providing multiple views within the same application
- The MVVM pattern can also allow code re-use in Universal application which share common logic and data but differ in presentation
  - E.g. Mobile vs Desktop views of the same data and controls

# Add a Map View Class

- Create a new file app/view/Map.js
- Extend 'Ext.Component'
- Methods for integrating with the Google Maps API library
- publishes: ['selection']
- updateSelection() method
- updateStore() method

```
Ext.define('Earthquakes.view.Map', {
    extend: 'Ext.Component',
    xtype: 'earthquakesmap',

    renderConfig: {
        selection: null,
        location: null,
        store: null
    },
    publishes: ['selection'],
    padding: 8,
    html: 'Please specify a location and marker coordinates.',

    applyStore: function(store) {
        return Ext.data.StoreManager.lookup(store);
    },
    updateStore: function(store) {
        var me = this;
        if (store.isLoaded()) {
            me.drawMarkers(store);
        } else {
            store.on('load', function(store) {
                me.drawMarkers(store);
            });
        }
    },
    getMap: function() {
        this.map = this.map || new google.maps.Map(this.getEl().dom, {
            zoom: 6,
            mapTypeId: google.maps.MapTypeId.TERRAIN
        });
        return this.map;
    },
    updateLocation: function(location) {
        if (location) {
            this.getMap().panTo({
                lat: location.latitude,
                lng: location.longitude
            });
        }
    },
    updateSelection: function(record, oldItem) {
        var me = this,
            markers = me.getMarkers();

        Ext.Array.forEach(markers, function(marker) {
            if (marker.record === record) {
                marker.setIcon(me.yellowIcon);
                marker.setZIndex(1000);
                me.fireEvent('select', this, record);
            } else if (marker.record === oldItem) {
                marker.setIcon(me.redIcon);
                marker.setZIndex(undefined);
            }
        });
    },
    // @private
    getMarkers: function() {
        return (this.markers || []);
    },
    // @private
    drawMarkers: function(store) {
        var me = this,
            markers = me.getMarkers(),
            map = me.getMap(),
            ll,
            marker,
            it;

        // Destroy the current set of markers.
        Ext.Array.forEach(markers, function(marker) {
            marker.setMap(null);
        });

        me.markers = [];

        // For each record, create and show a new marker, and push onto the array.
        store.each(function(record) {
            ll = new google.maps.LatLng(record.data.latitude, record.data.longitude);
            marker = new google.maps.Marker({
                position: ll,
                icon: me.redIcon,
                record: record
            });
            me.markers.push(marker);
            attachListener(marker, record);
            marker.setMap(map);
        });

        function attachListener(marker, record) {
            google.maps.event.addListener(marker, "click", function() {
                me.setSelection(record);
            });
        }
    },
    redIcon: {
        path: google.maps.SymbolPath.CIRCLE,
        scale: 10,
        strokeColor: 'black',
        strokeWeight: 3,
        fillColor: 'red',
        fillOpacity: 1.0
    },
    yellowIcon: {
        path: google.maps.SymbolPath.CIRCLE,
        scale: 10,
        strokeColor: 'black',
        strokeWeight: 3,
        fillColor: 'yellow',
        fillOpacity: 1.0
    }
});
```

# Adding an External Library

- Include Google Maps by adding <script/> tag to index.html

```
<script src="http://maps.googleapis.com/maps/api/js?  
libraries=geometry"></script>
```

- Google Maps API will require a Key for a published Web App

# Referencing the new Map View class

- In app/view/main/Main.js add class to *requires* array

```
requires: [ ... , "Earthquakes.view.Map"]
```

- Add Map View config to *items* array
- Linked to Map View by *xtype*
- split:true** allows user to adjust layout

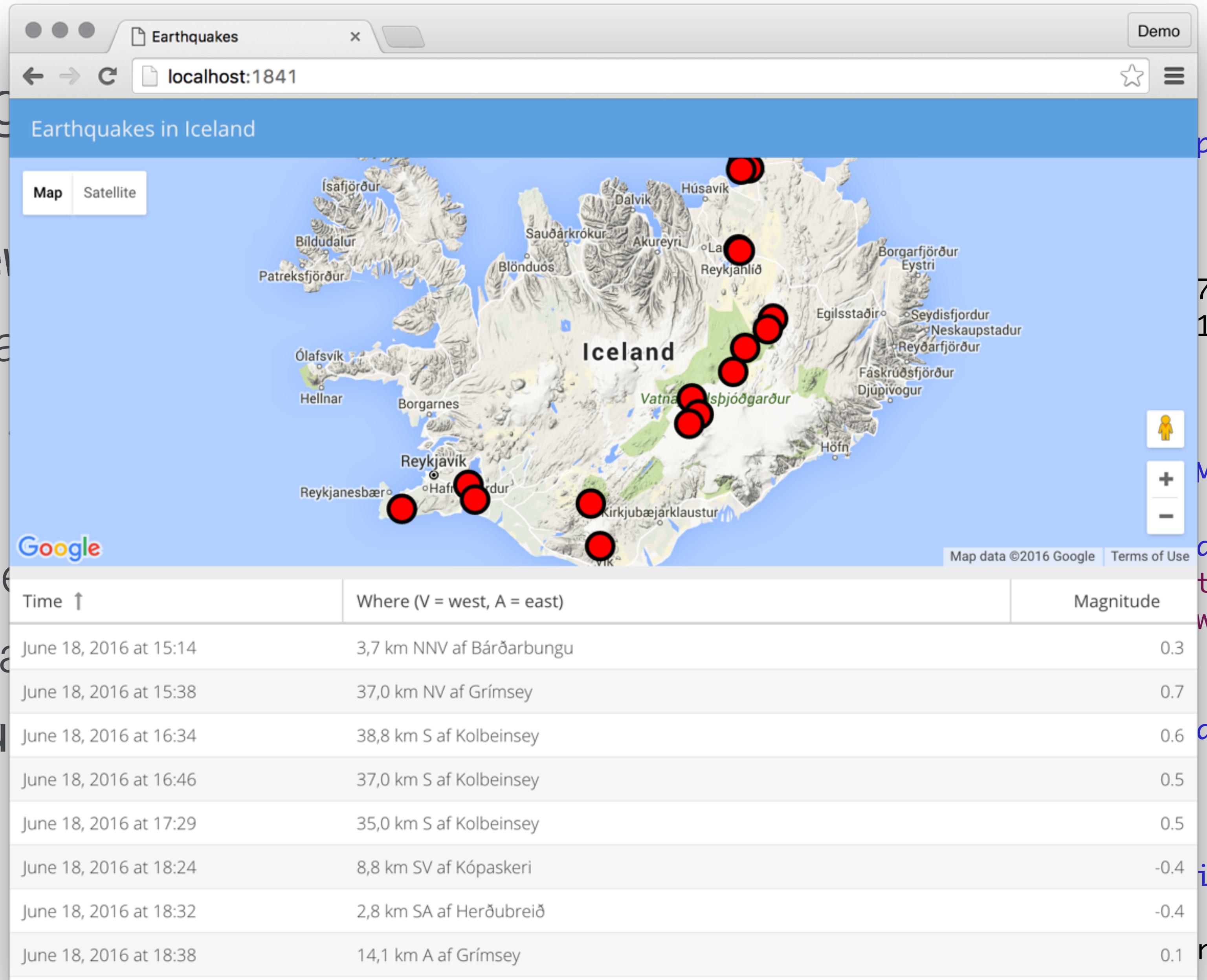
```
items: [{

}, {
    xtype: 'earthquakesmap',
    region: 'north',
    flex: 1,
    location: {
        latitude: 64.9312762,
        longitude: -19.021169
    },
    split: true,
    store: {
        model: 'Ext.data.Model',
        fields: [
            name: 'timestamp',
            convert: function(timestamp) {
                return new Date(timestamp);
            }
        ],
        sorters: ['timestamp'],
        autoLoad: true,
        proxy: {
            type: 'ajax',
            url: '//apis.is/earthquake/is',
            reader: {
                rootProperty: 'results'
            }
        }
    }
}]
```

# Referencing

- In app/view  
requires array  
requires: [

- Add Map View
- Linked to Model
- split:true



# Two Views, Two Stores, Two Ajax Requests

- Grid View and Map View both contain a Store
- The two Stores contain the same configuration
- Each Store will fetch, parse and manage the data independently
- Stores will not interact with each other

```
store: {  
    model: 'Ext.data.Model',  
    fields: [{  
        name: 'timestamp',  
        convert: function(timestamp) {  
            return new Date(timestamp);  
        }  
    }],  
    sorters: ['timestamp'],  
    autoLoad: true,  
    proxy: {  
        type: 'ajax',  
        url: '//apis.is/earthquake/is',  
        reader: {  
            rootProperty: 'results'  
        }  
    }  
}
```

# Shared Bindable Data

- One way to share the store between both the map and grid is to use a view model, which is a set of data that can be bound to the view
- Move store definition into a view model
- Bind store in the View

```
earthquakes: {  
    model: 'Ext.data.Model',  
    fields: [{  
        name: 'timestamp',  
        convert: function(timestamp) {  
            return new Date(timestamp);  
        }  
    }],  
    sorters: ['timestamp'],  
    autoLoad: true  
  
    items: [{  
        xtype: 'earthquakesmap',  
        bind: {  
            store: '{earthquakes}'  
        }  
    }]
```

# Create View Model Class

- Create the view model class file  
`app/view/main/MainModel.js`
- Add the new class to the *requires* array in  
`app/view/main/Main.js`  
`requires:[..., “Earthquakes.view.main.MainModel”]`
- Reference the view model from the view  
`viewModel: { type: ‘main’ },`
- Replace the store definitions with data binding in `app/view/main/Main.js`  
`bind: { store: ‘{earthquakes}’ }`

```
Ext.define('Earthquakes.view.main.MainModel', {
    extend: 'Ext.app.ViewModel',
    alias: 'viewmodel.main',
    data: {},
    formulas: {},
    stores: {
        earthquakes: {
            model: 'Ext.data.Model',
            fields: [
                {
                    name: 'timestamp',
                    convert: function(timestamp) {
                        return new Date(timestamp);
                    }
                }
            ],
            sorters: ['timestamp'],
            autoLoad: true,
            proxy: {
                type: 'ajax',
                url: '//apis.is/earthquake/is',
                reader: {
                    rootProperty: 'results'
                }
            }
        }
    }
});
```

# Shared Store

- Store now belongs to the view model
- Both Map and Grid views are bound to view model's *earthquakes* store
- Store data is fetched once and displayed twice
- Updating the store will update both views

# Handling Events

- In JavaScript events are handled by registering a listener function
- Different views may require different events to be handled
  - Mobile devices will listen for touch events
  - Desktop devices will listen for click events

# Add Map Selection Event

- Display the details of the selected map marker
- Add a *select listener* to `app/view/main/Main.js`
- Display an `Ext.Toast()` notification
- Need to add “`Ext.window.Toast`” to *requires array*

```
listeners: {
    select: function(map, record) {
        var data = record.data;
        var time = Ext.Date.format(data.timestamp, 'F j, g:i a');
        var s = 'A magnitude ' + data.size + ' earthquake occurred '
               + data.humanReadableLocation + '.';
        Ext.toast(s, time, 't'); // Message, title, alignment (top)
    }
}
```

# Add Map Selection

- Display the coordinates of a selected map marker
- Add a selection listener to app/view/
- Display an Earthquake table
- Need to add a listener to the table that requires array

The screenshot shows a web browser window with the title 'Earthquakes'. The URL is 'localhost:1841'. The page has a header 'Earthquakes in Iceland' and a timestamp 'June 19, 5:15 pm'. A tooltip on the map indicates a magnitude 0.7 earthquake occurred 18.1 km ASA af Trölladyngju. The map displays several red and yellow markers representing earthquake locations across Iceland. Below the map is a table with columns for Time, Where, and Magnitude.

Time ↑	Where (V = west, A = east)	Magnitude
June 19, 2016 at 13:52	4,3 km N af Herðubreiðarlindum	0.0
June 19, 2016 at 15:16	11,5 km V af Kópaskeri	0.4
June 19, 2016 at 16:12	2,9 km NNA af Hábungu	-0.2
June 19, 2016 at 17:15	18,1 km ASA af Trölladyngju	0.7
June 20, 2016 at 01:12	12,6 km A af Grímsey	0.4
June 20, 2016 at 02:37	12,8 km VSV af Kópaskeri	0.5
June 20, 2016 at 03:12	3,7 km SSV af Hrómundartindi	1.0
June 20, 2016 at 04:54	6,5 km SSA af Hveragerði	-1.0

# Separation of Concerns

- In MVVM the goal is to separate the view and the logic
- Move processing logic into a new view controller
- Move 'Ext.window.Toast' requirement to ViewController
- Link the view controller with the alias
- Add the new controller to the *requires* array in the Main view

## app/view/main/MainController.js

```
Ext.define('Earthquakes.view.main.MainController', {
    extend: 'Ext.app.ViewController',
    alias: 'controller.main',
    requires: ['Ext.window.Toast'],

    onSelect: function(map, record) {
        var data = record.data;
        var time = Ext.Date.format(data.timestamp, 'F j, g:i a');
        var s = 'A magnitude ' + data.size + ' earthquake occurred '
            + data.humanReadableLocation + '.';
        Ext.toast(s, time, 't');
    }
});
```

## app/view/main/Main.js

```
viewController: {
    type: 'main'
},
listeners: {
    select: 'onSelect'
},
```

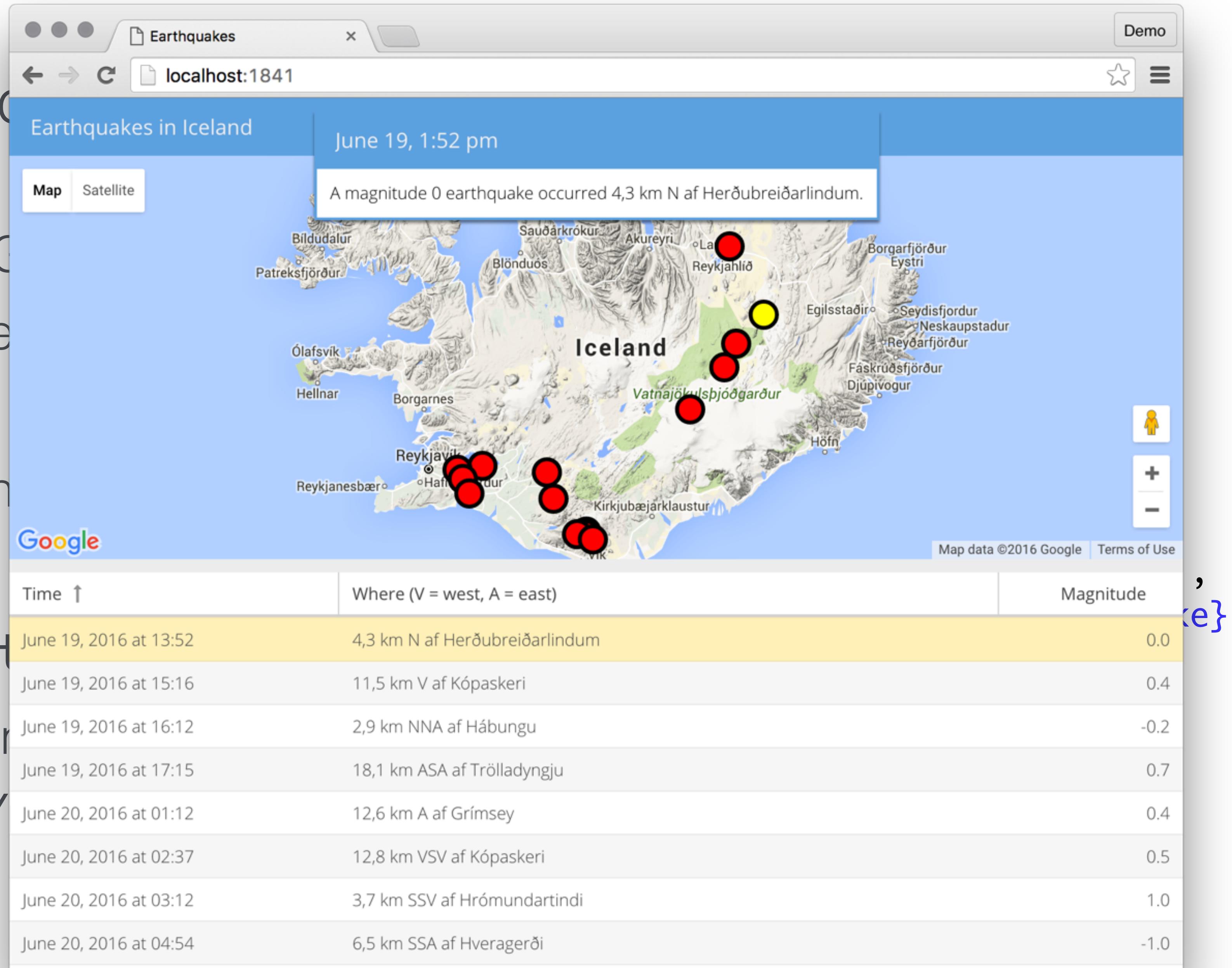
# Update Grid Selection with Map Selection

- Since both Grid and Map publish their selection we can bind these values
- Changes to either value will then be passed to the `setSelection()` method
- Add `selection` to both bind configurations (for Map and Grid) in `app/view/main/Main.js`

```
bind: {  
    store: '{earthquakes}',  
    selection: '{earthquake}'  
}
```

# Update Grid

- Since both Controller and Model have selection we can pass changes to the grid
- Changes to selection are passed to the grid's selectionChanged method
- Add selectionChanged configuration to app/view/Earthquakes

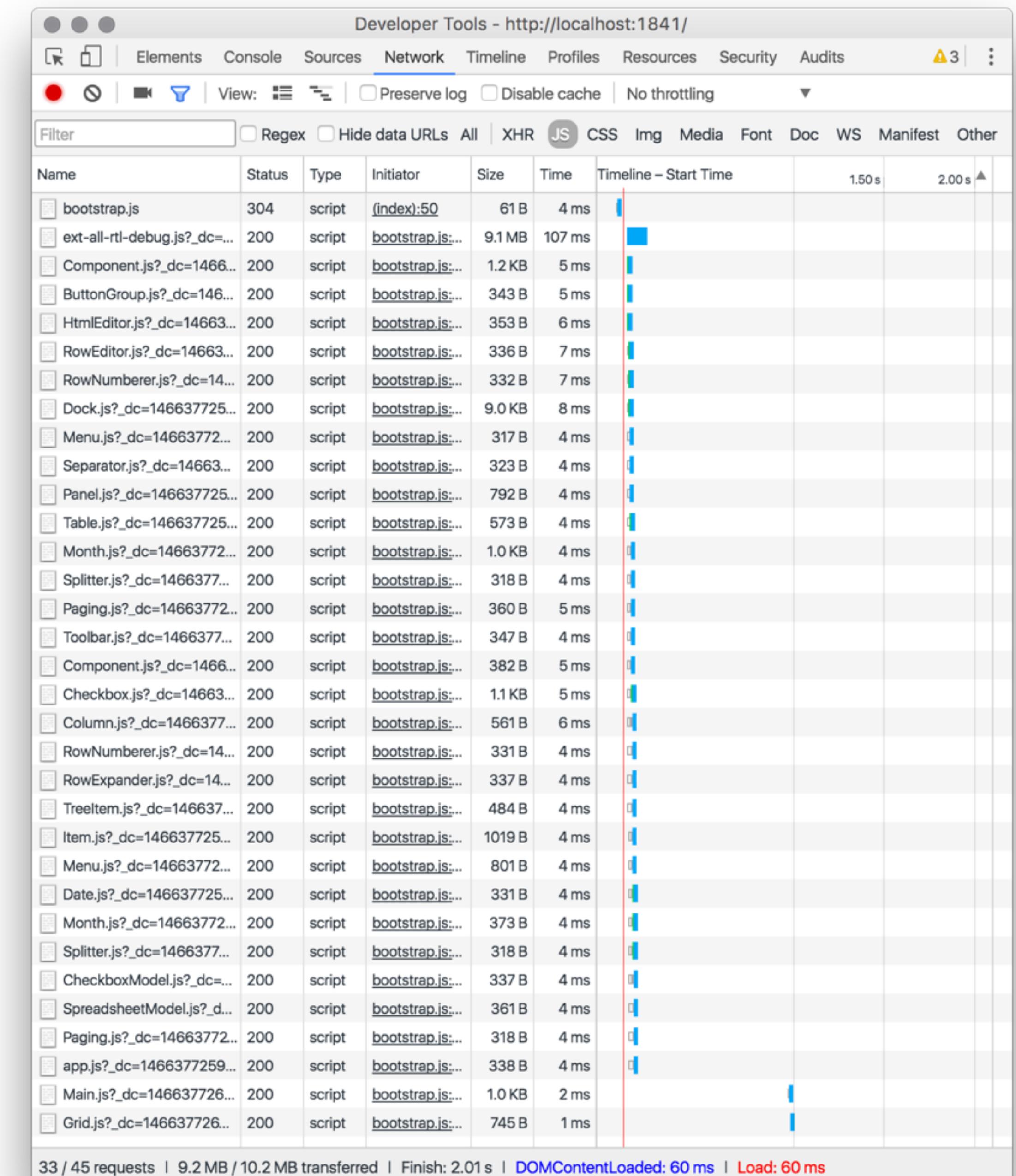


# Optimisation

Building for production

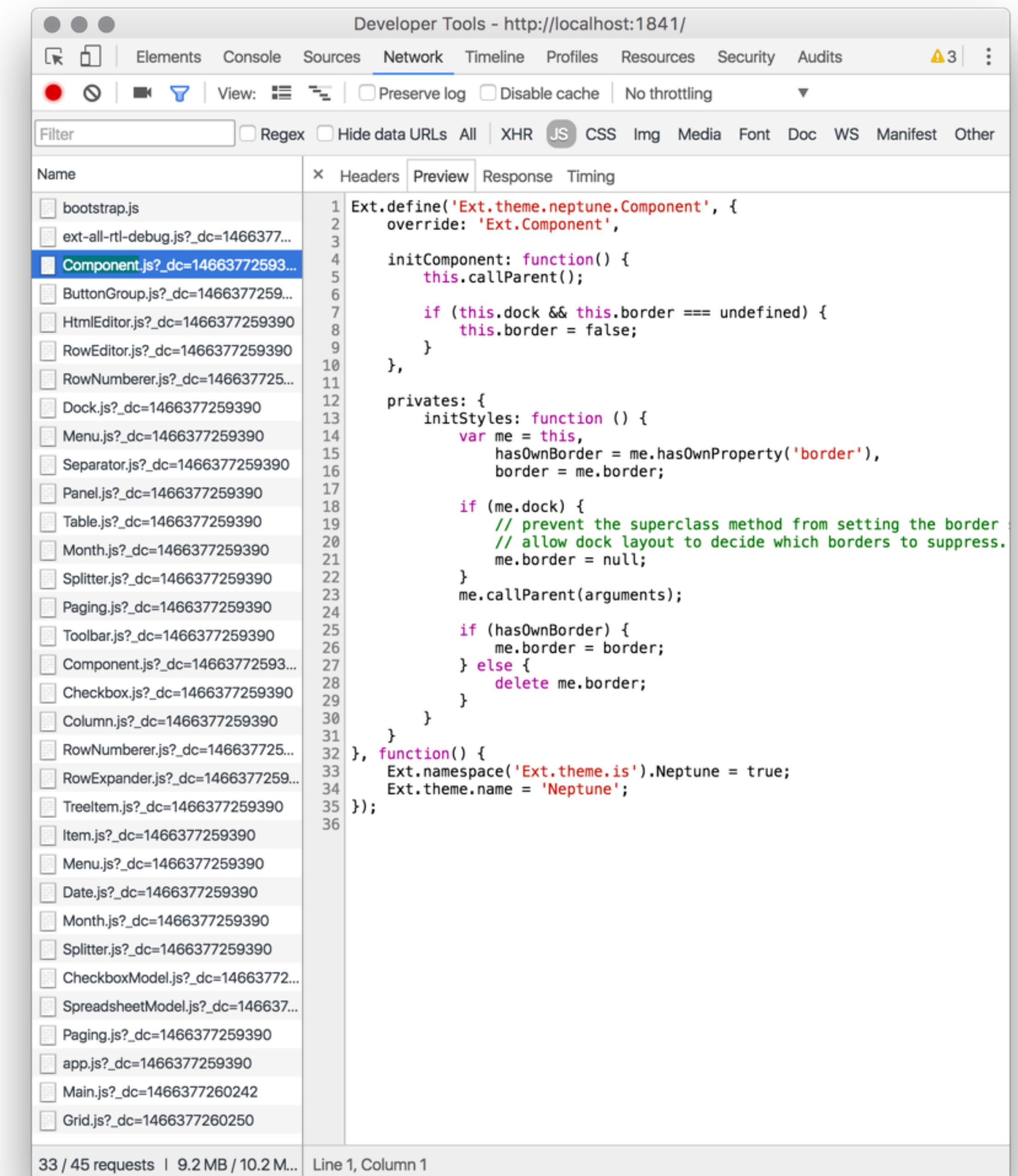
# Development Network Traffic

- We can view network traffic for our development build
- Each class file is loaded separately
- 33 JavaScript files are requested
- 9.2 MB of JavaScript is transferred



# Development Build Files

- Load order managed by *bootstrap.js*
- Each class file is loaded in the original format containing comments and whitespace
- Timestamp added to prevent unwanted caching in development



The screenshot shows the Network tab in Google Chrome's Developer Tools. A list of files is displayed on the left, and the content of the selected file, 'Component.js?\_dc=14663772593...', is shown on the right. The file contains ExtJS code defining a component named 'Ext.theme.neptune.Component'.

```
Ext.define('Ext.theme.neptune.Component', {
    override: 'Ext.Component',
    initComponent: function() {
        this.callParent();
        if (this.dock && this.border === undefined) {
            this.border = false;
        }
    },
    privates: {
        initStyles: function () {
            var me = this,
                hasOwnBorder = me.hasOwnProperty('border'),
                border = me.border;
            if (me.dock) {
                // prevent the superclass method from setting the border
                // allow dock layout to decide which borders to suppress.
                me.border = null;
            }
            me.callParent(arguments);
            if (hasOwnBorder) {
                me.border = border;
            } else {
                delete me.border;
            }
        }
    },
    function() {
        Ext.namespace('Ext.theme.is').Neptune = true;
        Ext.theme.name = 'Neptune';
    });
});
```

At the bottom of the developer tools window, it says '33 / 45 requests | 9.2 MB / 10.2 M...' and 'Line 1, Column 1'.

# Back to the Build Tool

The build tool should manage

- Load Order - Ensure dependencies are load prior to being used
- Optimisation - Pre-process scripts for improved performance
- Minimisation - Reduce the amount of data being loaded by removing comments and whitespace
- Obfuscation - Replace descriptive variable and function names with short ones
- Concatenation - Bundle multiple files together to reduce network requests
- Caching - Store files in browser for improved loading on future visits

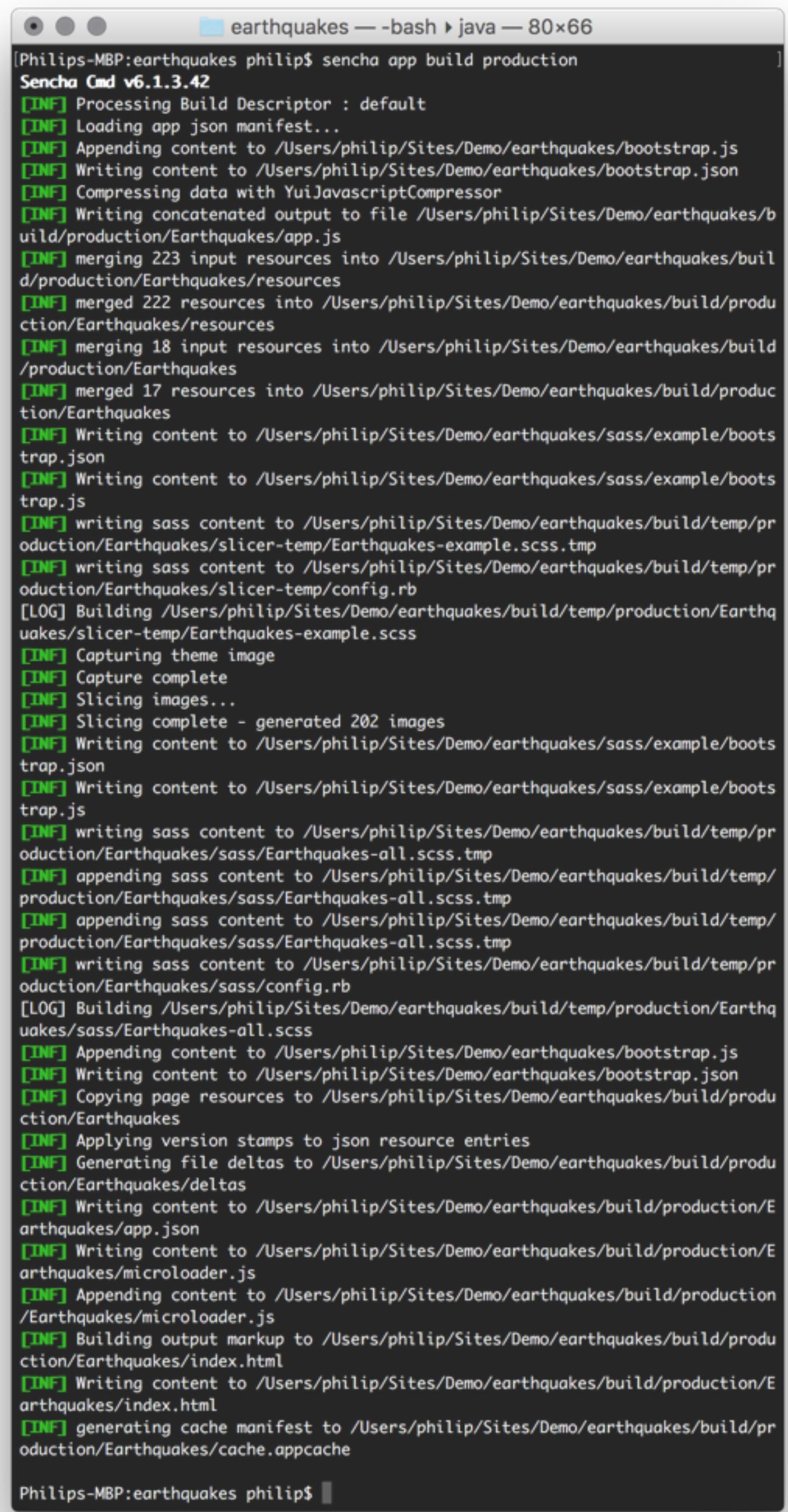
# Sencha Cmd for Production

- Generating a production build with Sencha Cmd

`sencha app build production`

- Production Web App is generated in the build/production directory
- Load the production build at

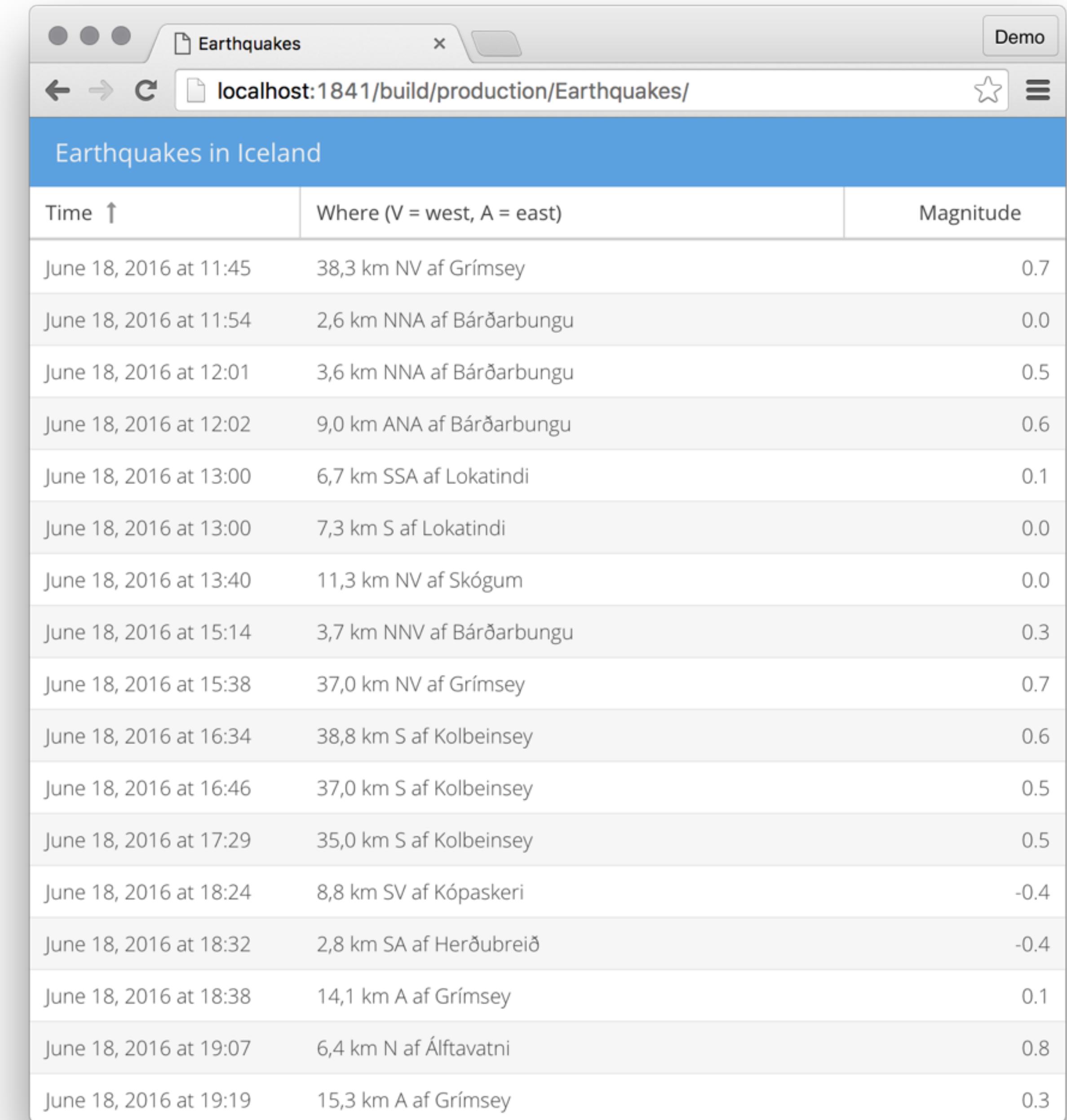
`http://localhost:1841/build/production/Earthquakes/`



```
[Philip-MBP:earthquakes philip$ sencha app build production
Sencha Cmd v6.1.3.4
[INF] Processing Build Descriptor : default
[INF] Loading app.json manifest...
[INF] Appending content to /Users/philip/Sites/Demo/earthquakes/bootstrap.js
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/bootstrap.json
[INF] Compressing data with YuiJavascriptCompressor
[INF] Writing concatenated output to file /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/app.js
[INF] merging 223 input resources into /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/resources
[INF] merged 222 resources into /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/resources
[INF] merging 18 input resources into /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes
[INF] merged 17 resources into /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.json
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.js
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/slicer-temp/Earthquakes-example.scss.tmp
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/slicer-temp/config.rb
[LOG] Building /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/slicer-temp/Earthquakes-example.scss
[INF] Capturing theme image
[INF] Capture complete
[INF] Slicing images...
[INF] Slicing complete - generated 202 images
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.json
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/sass/example/bootstrap.js
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] appending sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] appending sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/sass/Earthquakes-all.scss.tmp
[INF] writing sass content to /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/sass/config.rb
[LOG] Building /Users/philip/Sites/Demo/earthquakes/build/temp/production/Earthquakes/sass/Earthquakes-all.scss
[INF] Appending content to /Users/philip/Sites/Demo/earthquakes/bootstrap.js
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/bootstrap.json
[INF] Copying page resources to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes
[INF] Applying version stamps to json resource entries
[INF] Generating file deltas to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/deltas
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/app.json
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/microloader.js
[INF] Appending content to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/microloader.js
[INF] Building output markup to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/index.html
[INF] Writing content to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/index.html
[INF] generating cache manifest to /Users/philip/Sites/Demo/earthquakes/build/production/Earthquakes/cache.appcache
Philip-MBP:earthquakes philip$ ]
```

# Production Build Files

- Visually identical to development build
- JavaScript source has been
  - Loaded in correct order
  - Minimised
  - Obfuscated
  - Concatenated

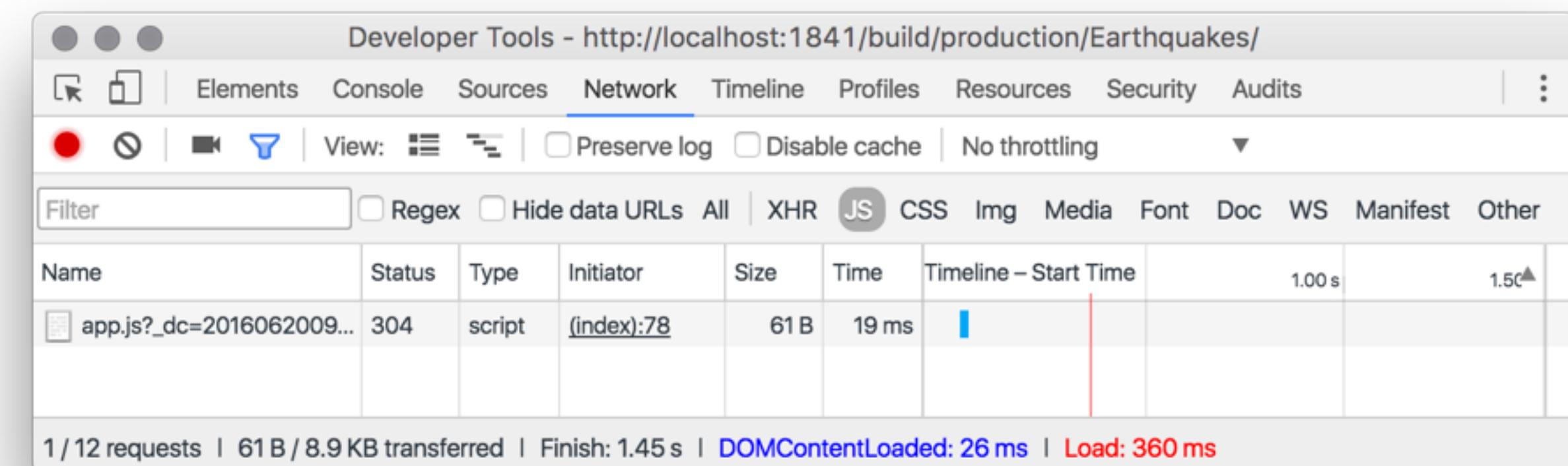
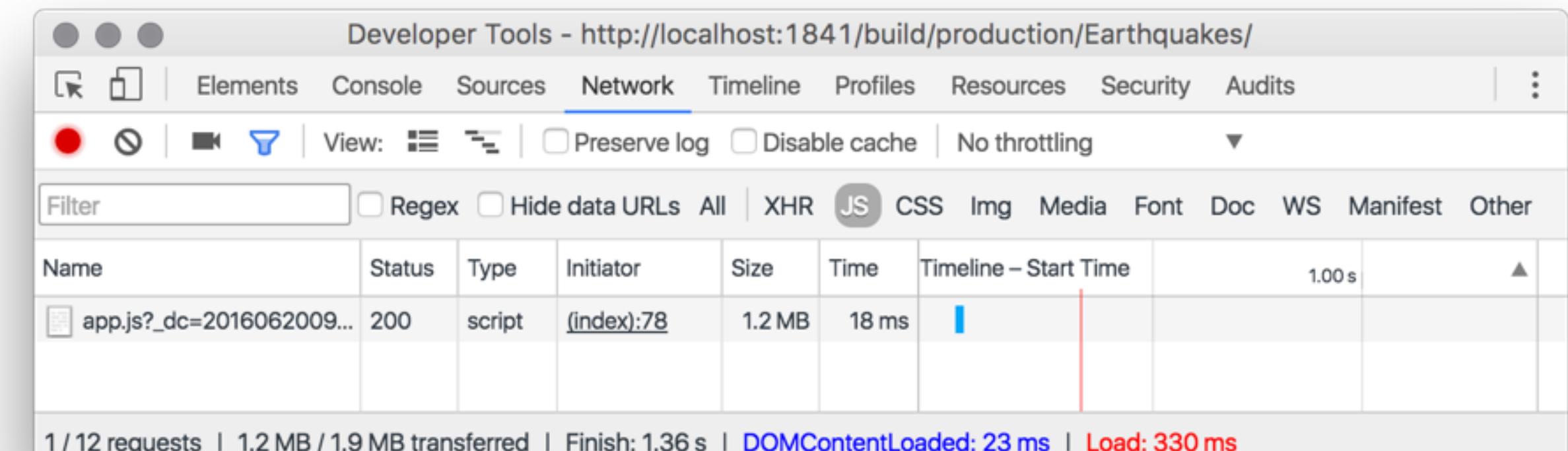


The screenshot shows a web browser window with the title bar "Earthquakes" and the URL "localhost:1841/build/production/Earthquakes/". The page content is titled "Earthquakes in Iceland" and displays a table of 15 rows of earthquake data. The columns are "Time ↑", "Where (V = west, A = east)", and "Magnitude". The data is as follows:

Time ↑	Where (V = west, A = east)	Magnitude
June 18, 2016 at 11:45	38,3 km NV af Grímsey	0.7
June 18, 2016 at 11:54	2,6 km NNA af Bárðarbungu	0.0
June 18, 2016 at 12:01	3,6 km NNA af Bárðarbungu	0.5
June 18, 2016 at 12:02	9,0 km ANA af Bárðarbungu	0.6
June 18, 2016 at 13:00	6,7 km SSA af Lokatindi	0.1
June 18, 2016 at 13:00	7,3 km S af Lokatindi	0.0
June 18, 2016 at 13:40	11,3 km NV af Skógum	0.0
June 18, 2016 at 15:14	3,7 km NNV af Bárðarbungu	0.3
June 18, 2016 at 15:38	37,0 km NV af Grímsey	0.7
June 18, 2016 at 16:34	38,8 km S af Kolbeinsey	0.6
June 18, 2016 at 16:46	37,0 km S af Kolbeinsey	0.5
June 18, 2016 at 17:29	35,0 km S af Kolbeinsey	0.5
June 18, 2016 at 18:24	8,8 km SV af Kópaskeri	-0.4
June 18, 2016 at 18:32	2,8 km SA af Herðubreið	-0.4
June 18, 2016 at 18:38	14,1 km A af Grímsey	0.1
June 18, 2016 at 19:07	6,4 km N af Álfvatn	0.8
June 18, 2016 at 19:19	15,3 km A af Grímsey	0.3

# Production Network Traffic

- Microloader moved inline into index.html
- 1 JavaScript file loaded
- 1.2 MB of JavaScript transferred
- JavaScript is cached in browser
- Subsequently loaded from cache



# Web App Publication

- Publication build files can be moved into any web host
  - IIS
  - Apache
  - Node
  - Etc
- Build files can also be wrapped in native container
  - Electron for Desktop
  - Cordova / PhoneGap for Mobile

