

DIWICON Silver Cloud 6

Web Client - Developer Documentation

2014-05-19 Szilárd Szakács

Driving philosophy

1. SC6 is the web surface for our gas managemet service.
2. All reports are available in the web client as pages.
3. Report data is prepared by sc6 services.
4. The API serves standard Resources.
5. The web client is a Single Page Application.
6. All Resources have standard visual representations.
7. Do not show data not supported by user needs.
8. Show as much data as you can if it is needed.
9. The client is heavily convention based.
10. The UI is responsive to different device sizes.

Architecture

OVERVIEW

Presentation

HTML5 Client

Domain

Distributed Services
Application Services
Infrastructure

Infrastructure

Master
Data
Database

Master
Data
Database Data Access Layer
Application Cache Workflow
Persist
Databases

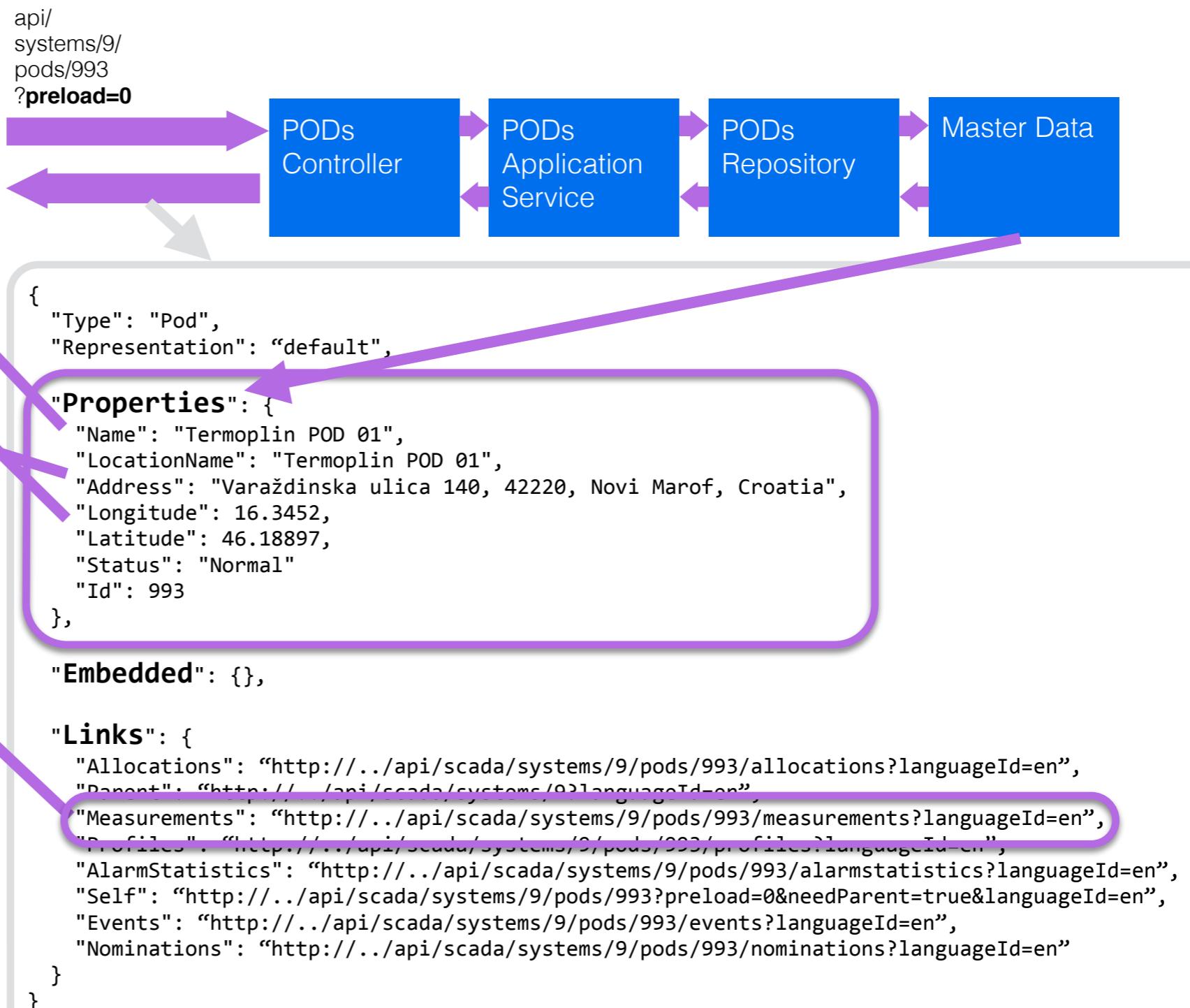
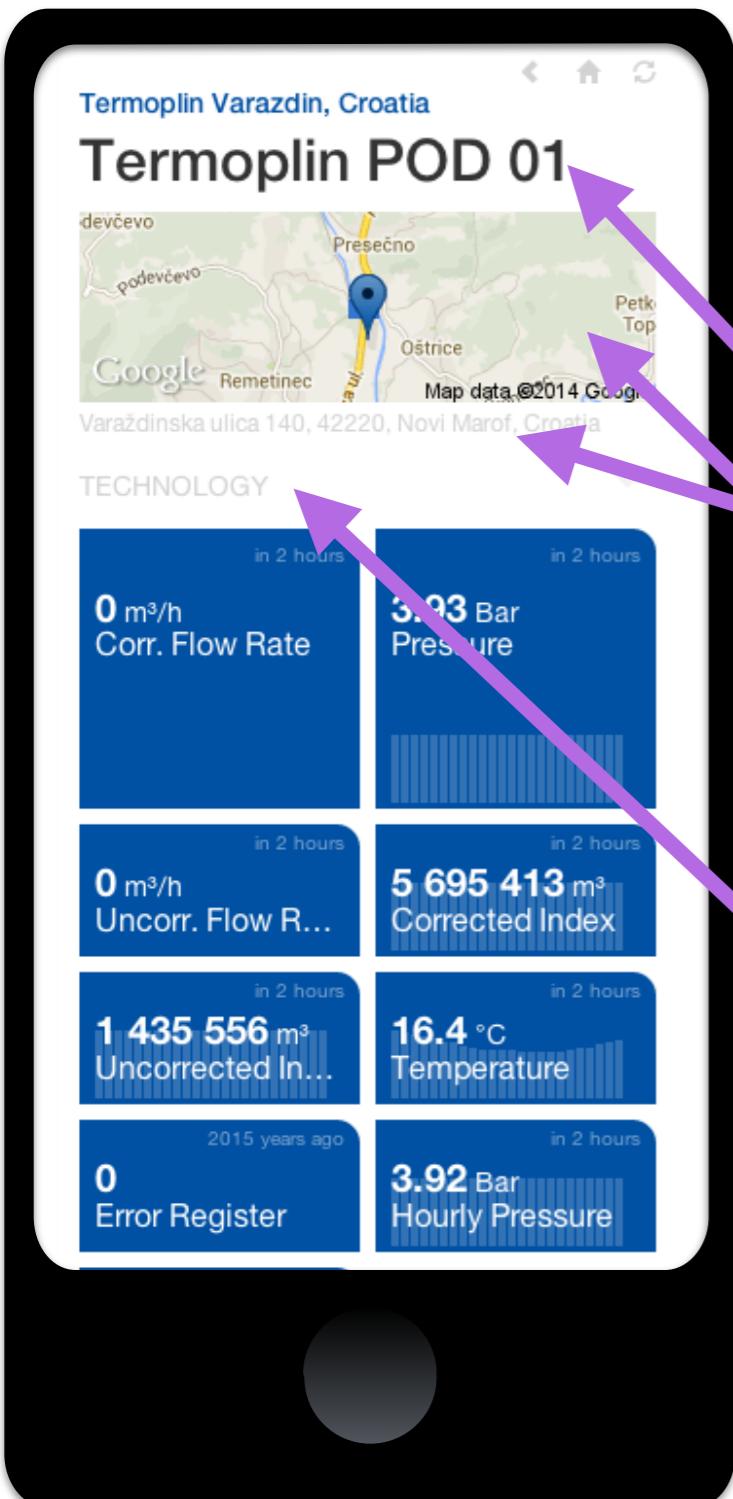
Workflows

Workflows

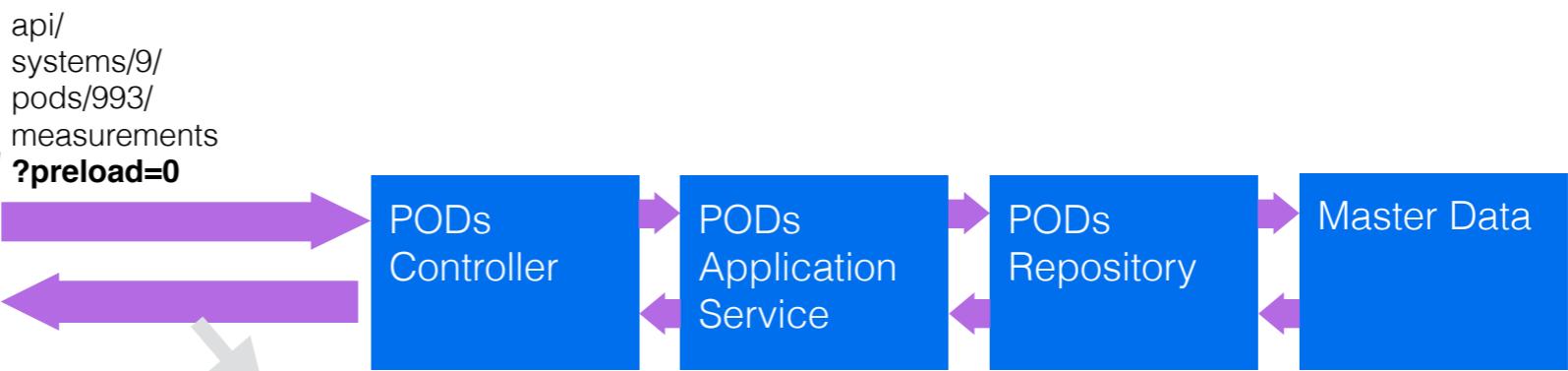
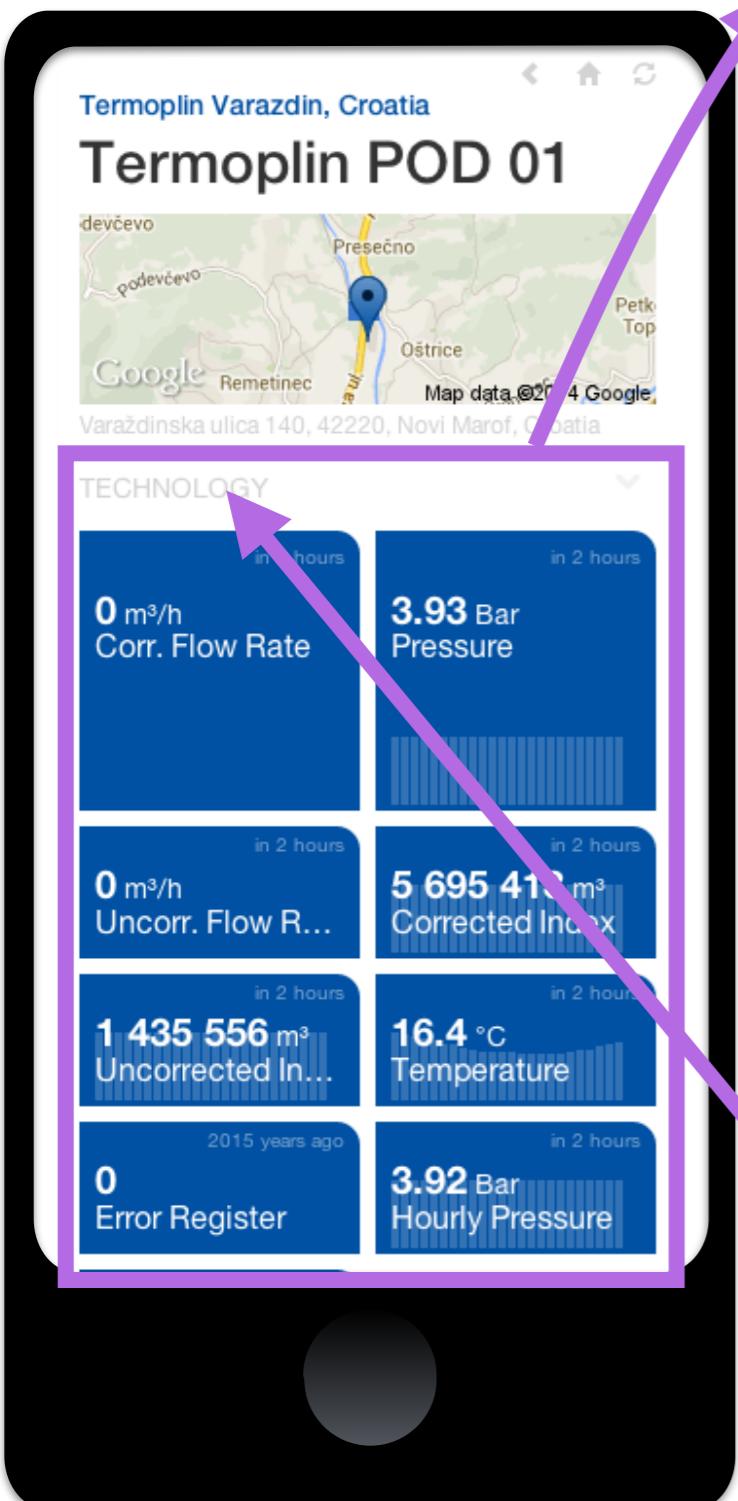
SC5

SC6

#/systems/9/pods/993

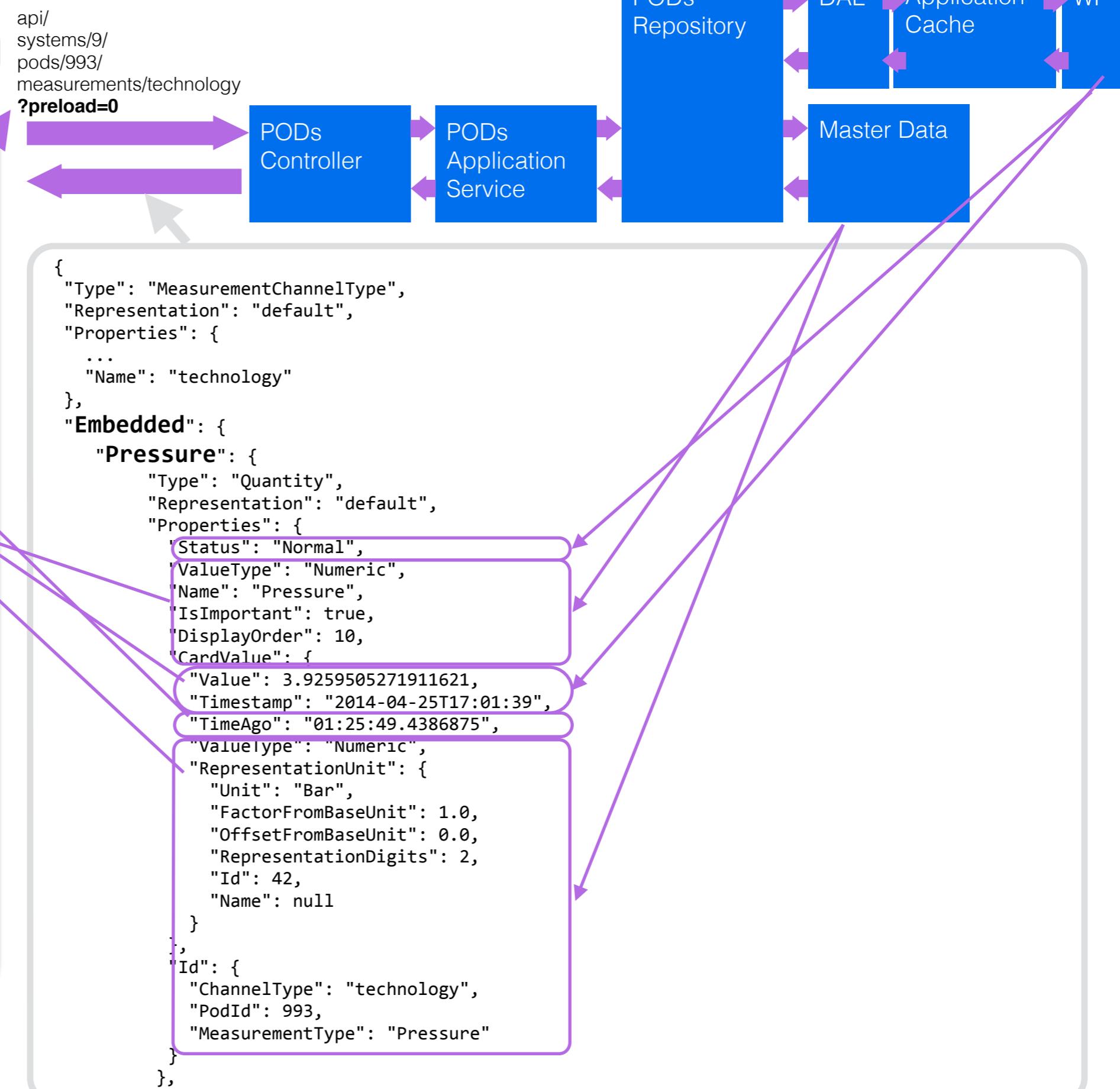
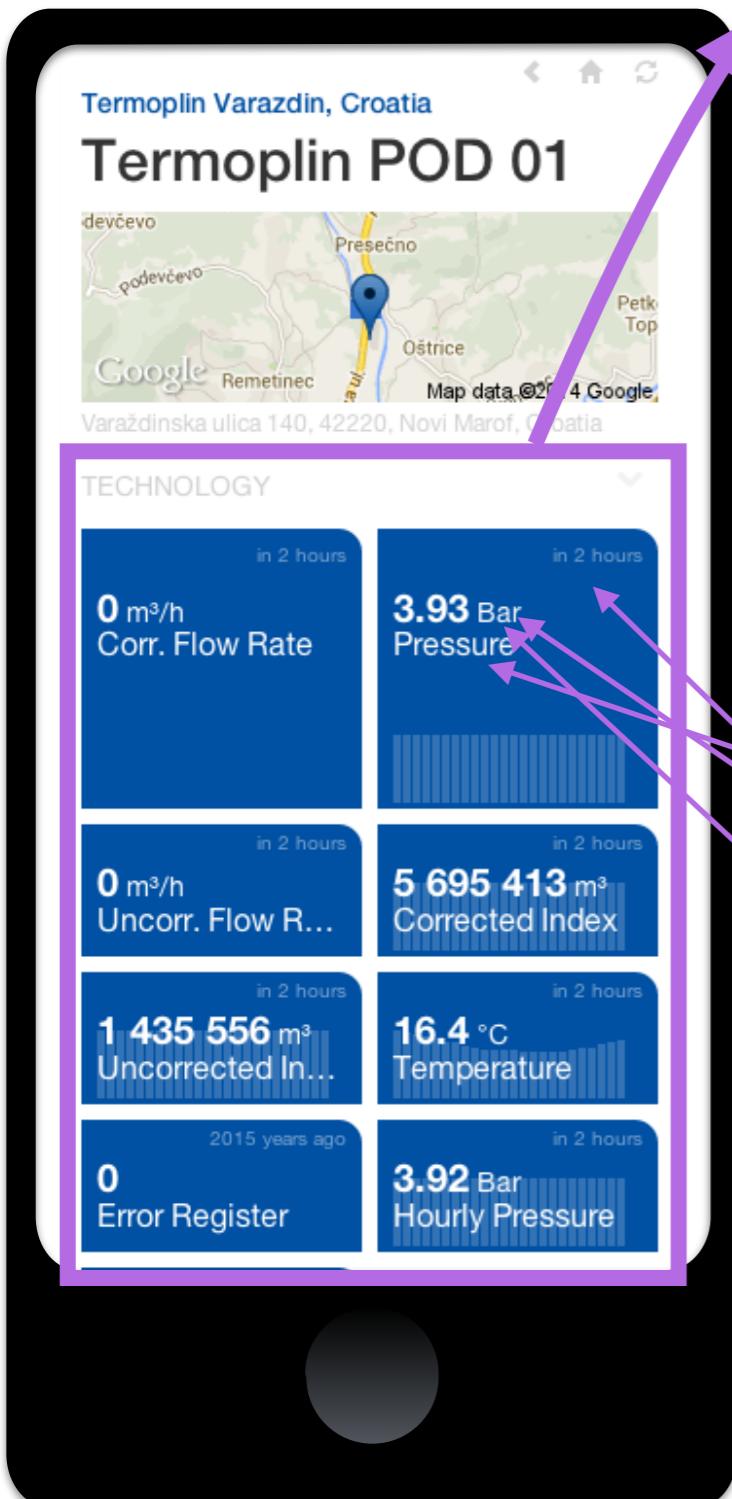


#/systems/9/pods/993

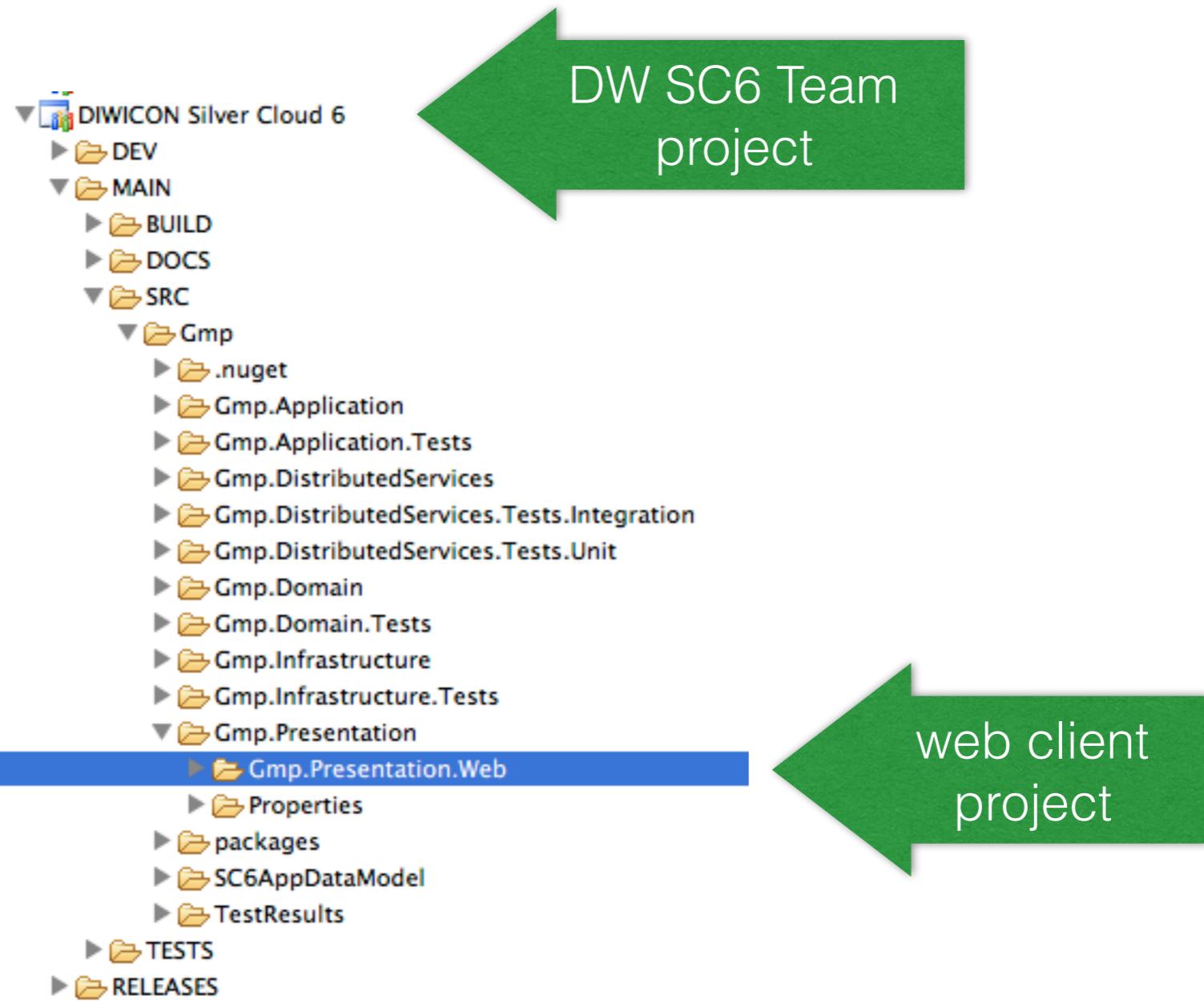


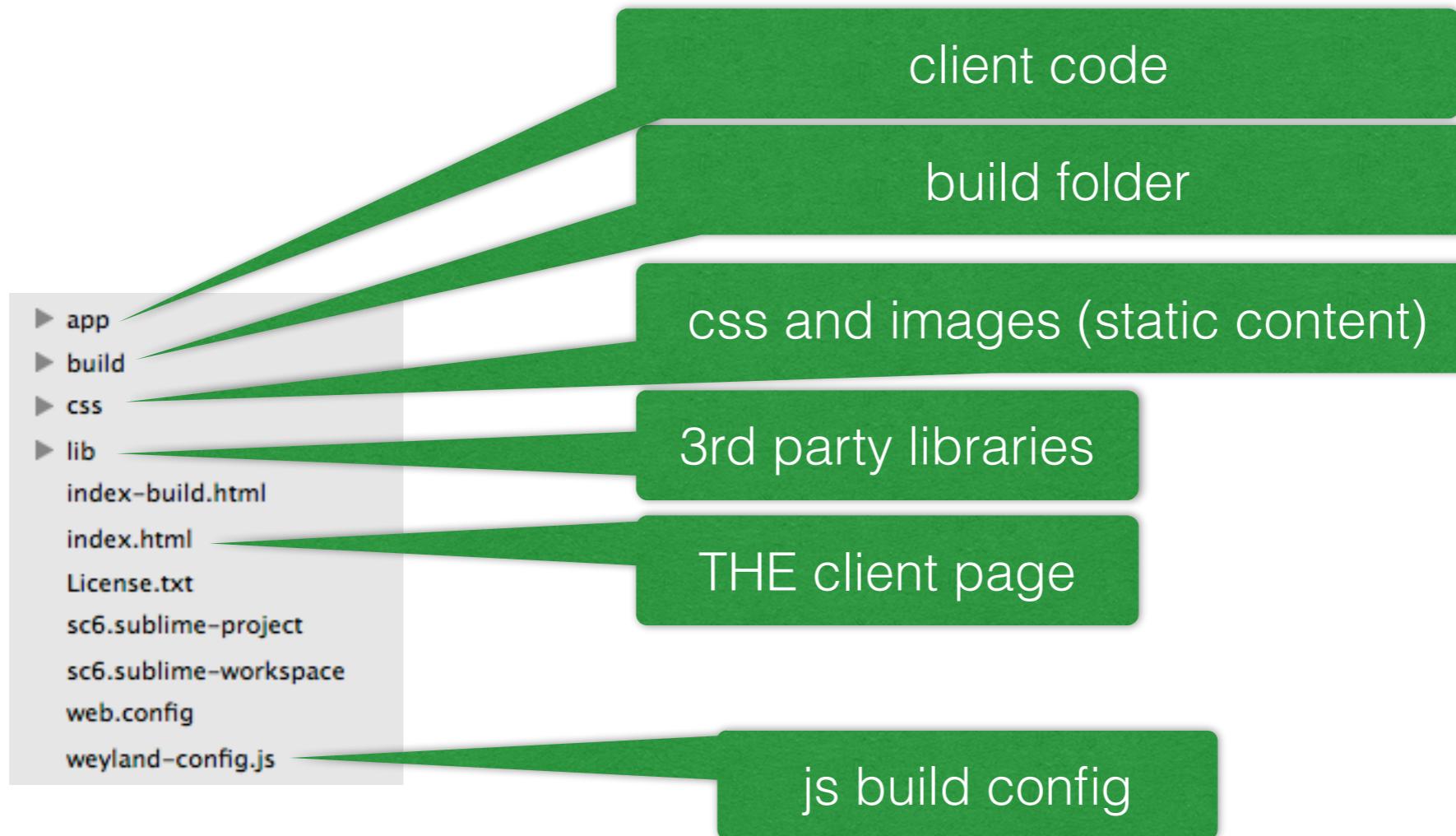
```
{  
    "Type": "Collection",  
    "Representation": "default",  
  
    "Properties": {  
        "Name": "MeasurementChannelTypes",  
        ...  
        "Take": 0  
    },  
  
    "Embedded": {  
    },  
  
    "Links": {  
        "Parent": "http://.../api/scada/systems/9/pods/993",  
        "Status": "http://.../api/scada/systems/9/pods/993/measurements/status",  
        "Communication": "http://.../api/scada/systems/9/pods/993/measurements/communication",  
        "Technology": "http://.../api/scada/systems/9/pods/993/measurements/technology",  
        "Self": "http://.../api/scada/systems/9/pods/993/measurements"  
    }  
}
```

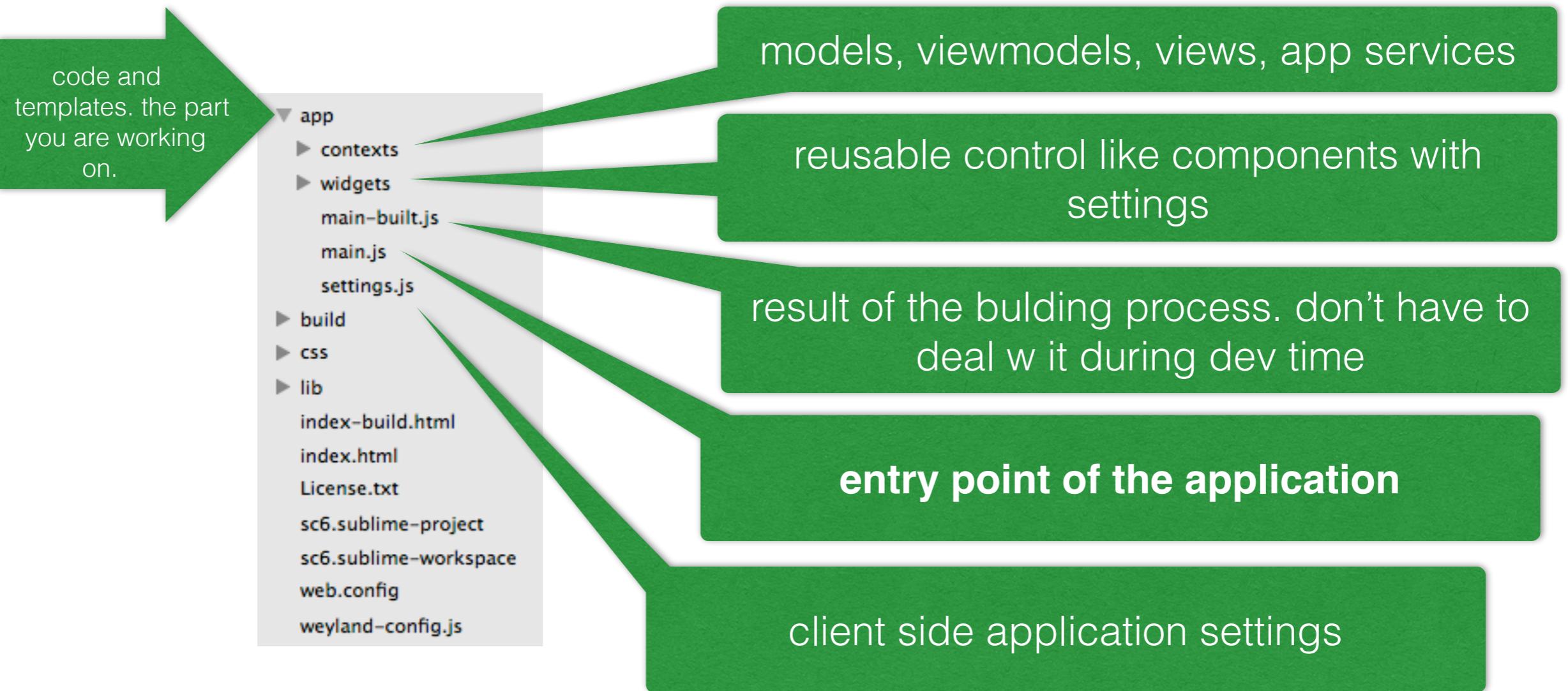
#/systems/9/pods/993

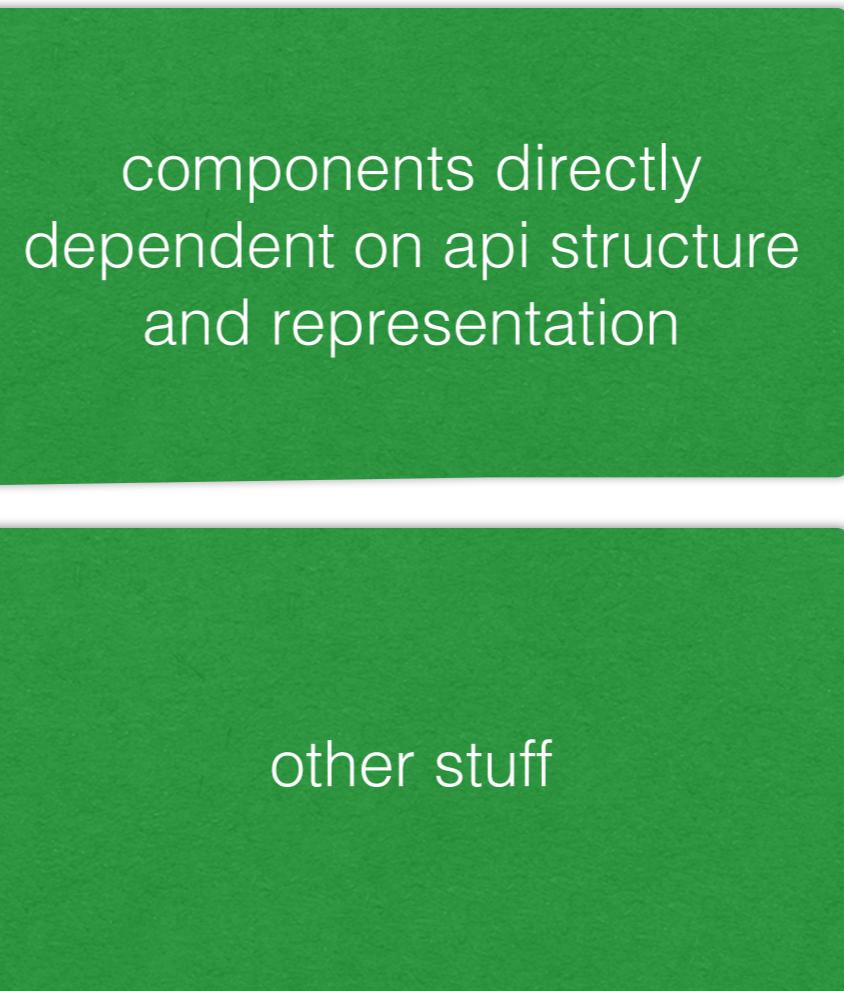
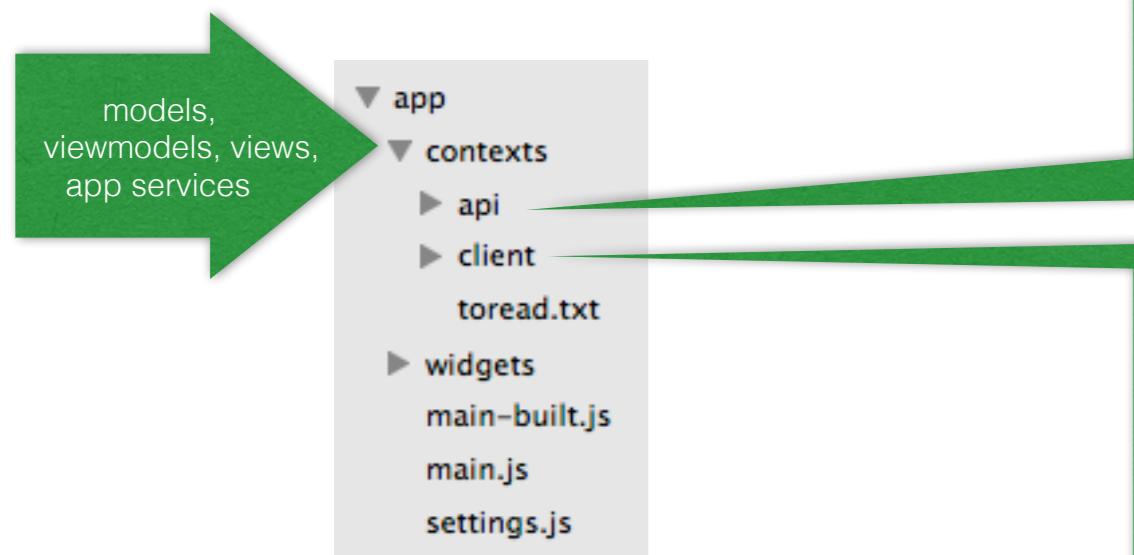


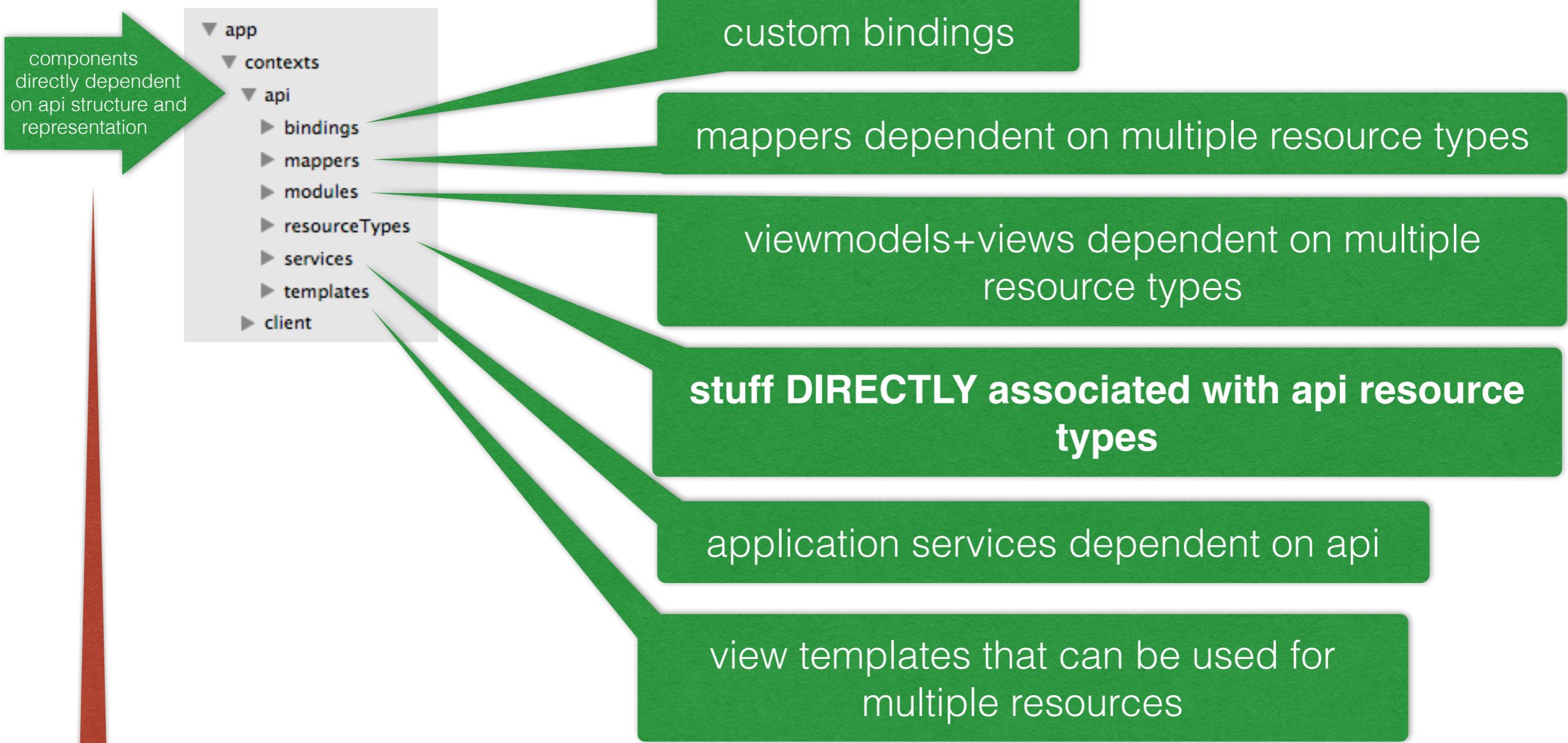
Project structure



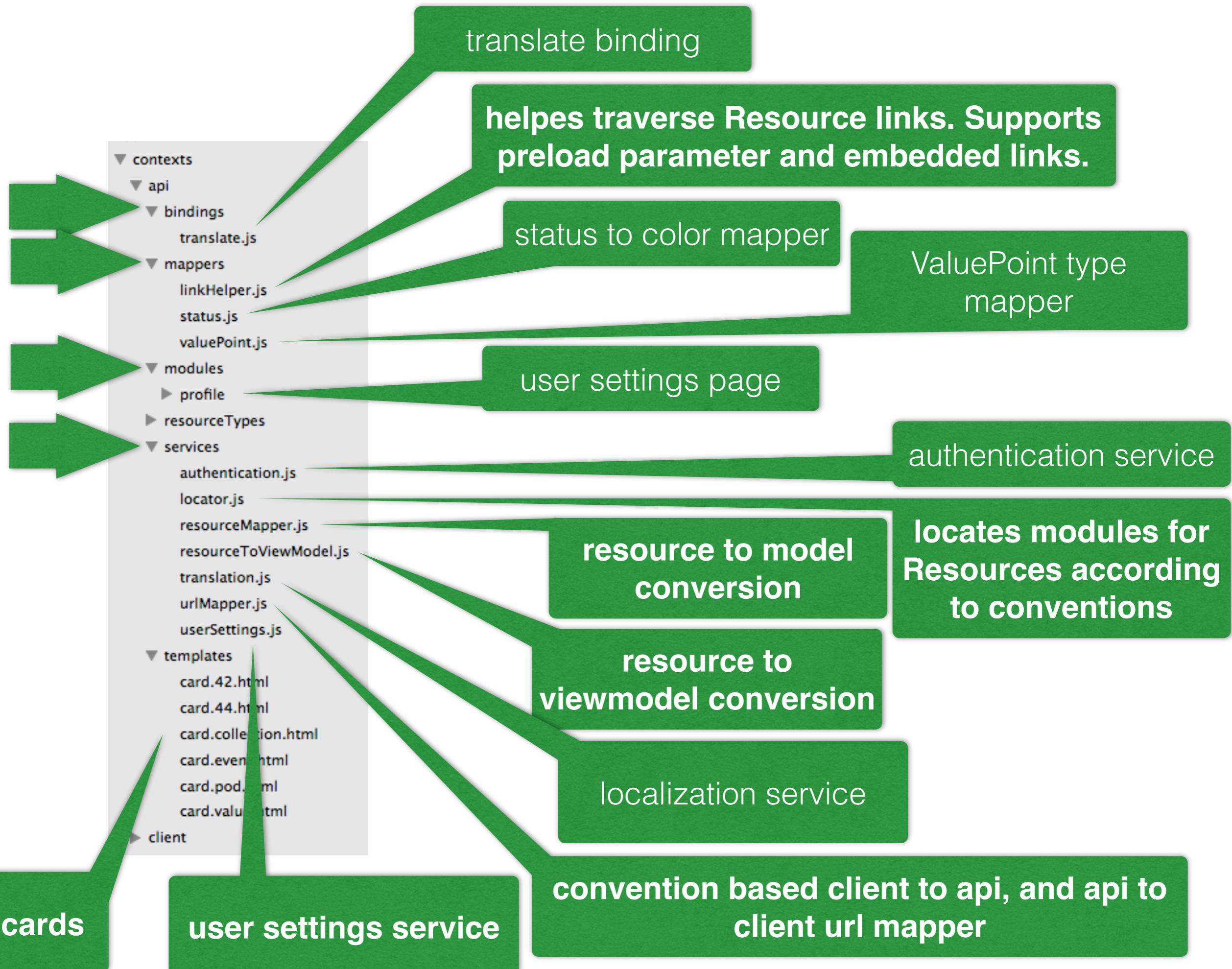


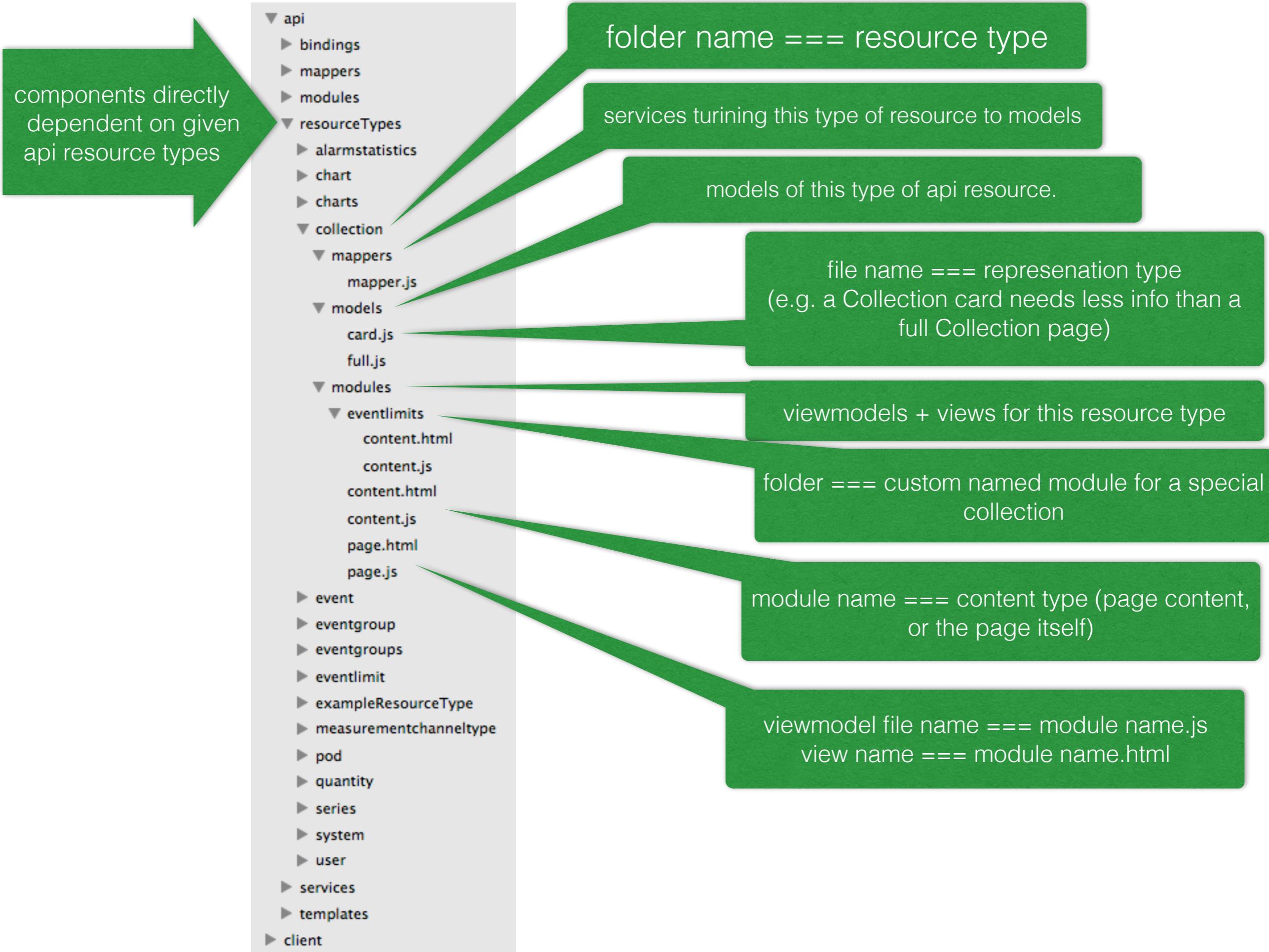






!!! structure is important here as it fuels many convention based mechanisms !!!





folder name === resource type

services turning this type of resource to models

models of this type of api resource.

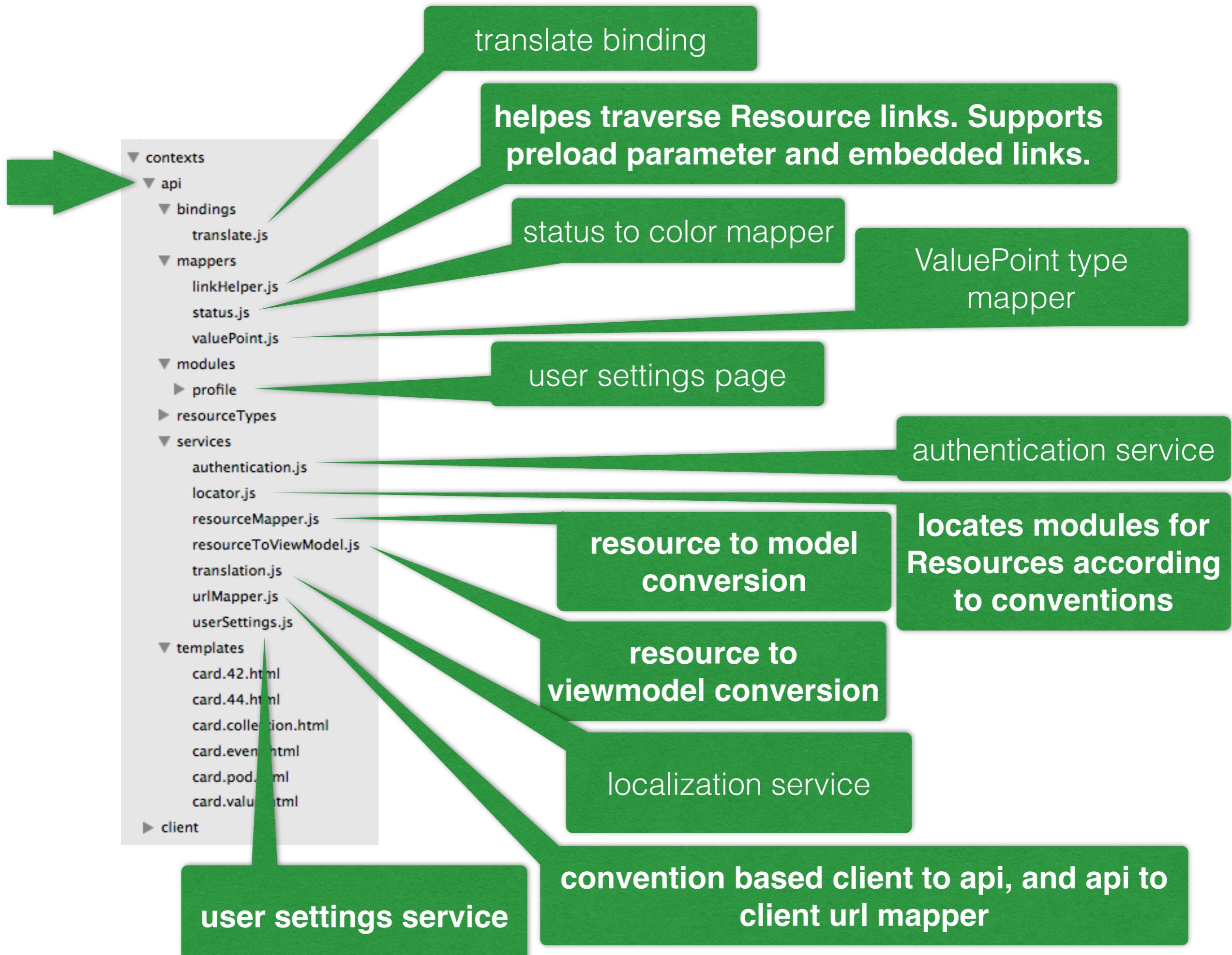
file name === representation type
(e.g. a Collection card needs less info than a full Collection page)

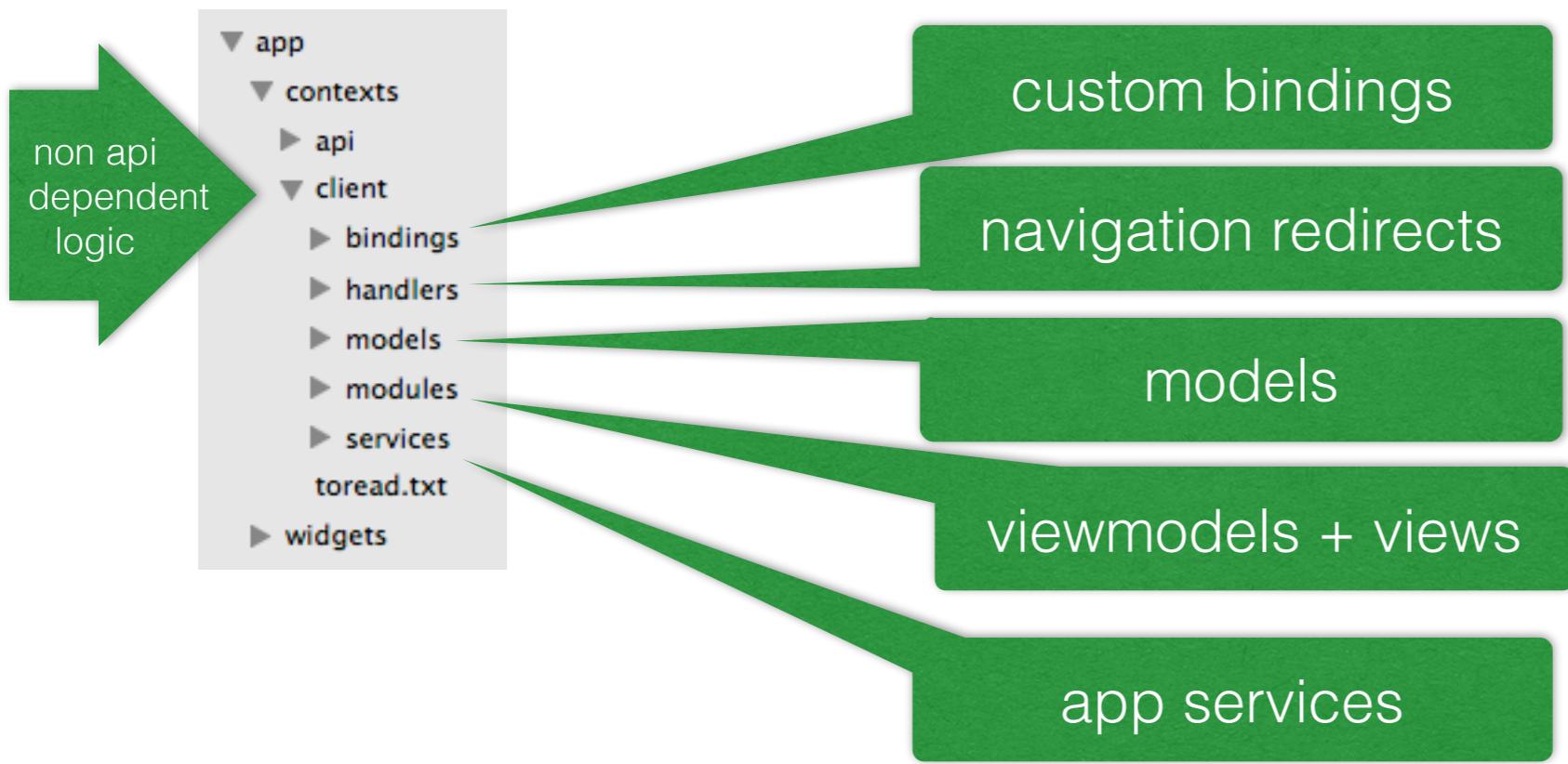
viewmodels + views for this resource type

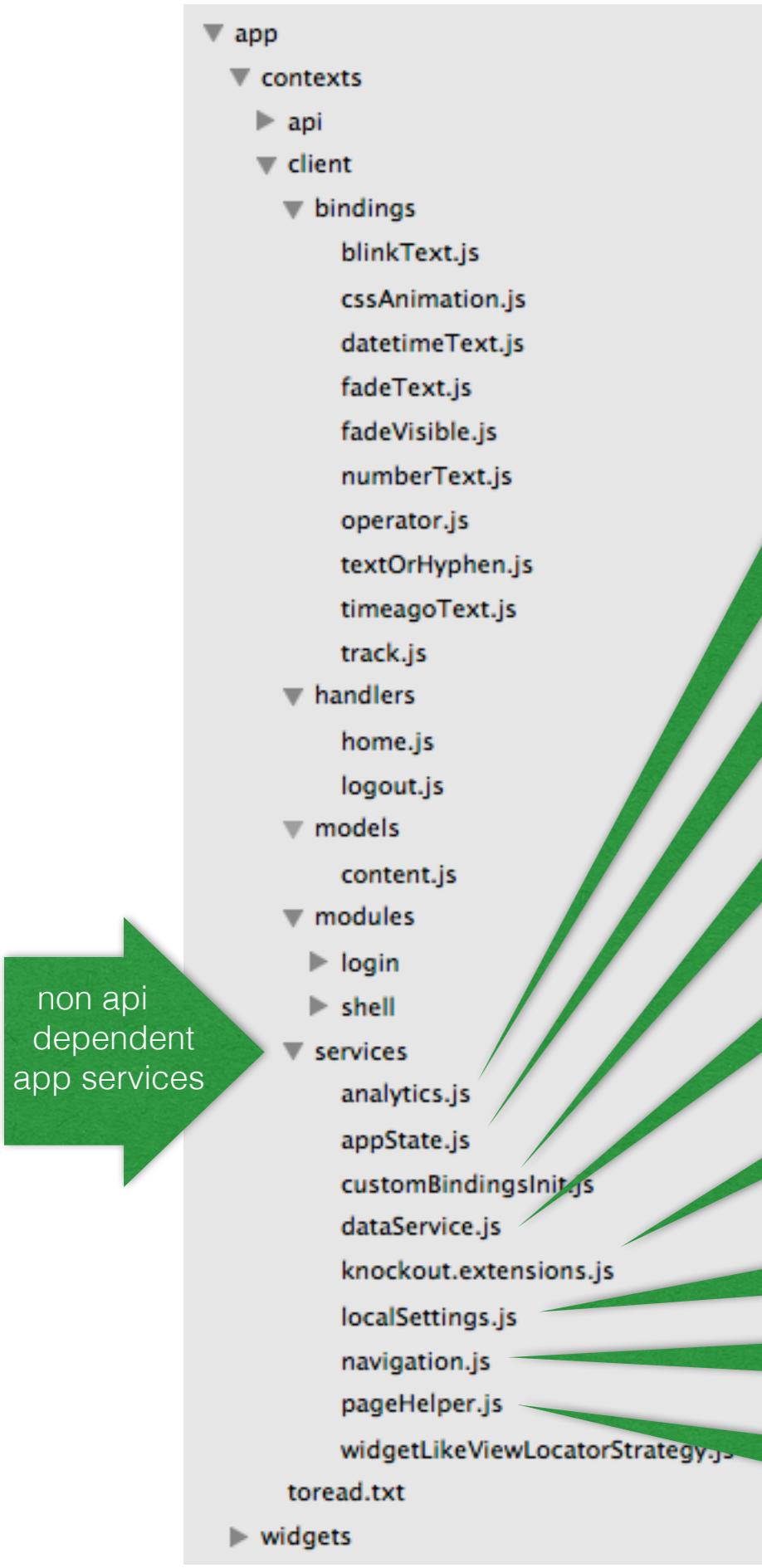
folder === custom named module for a special collection

module name === content type (page content, or the page itself)

viewmodel file name === module name.js
view name === module name.html







user behavior analytics (google analytics wrapper)

saves app state between sessions (not completed)

loads custom binding modules

http service w api specific support (should be in api context)

dirty flag support for knockout

settings service for local settings (local storage, time to live support)

convention based parent link generator for api resources (unused)

updates page title after navigation

renders content setting. remembers last size.
animated to new size.

reusable control
like components with
settings

```
▼ app
  ► contexts
  ▼ widgets
    ► animatedcompositionhost
    ► datepicker
    ► expander
    ► map
    ► minitrend
    ► packeryarea
    ► parentlink
    ► seriesviewer
    ► spinner
    ► togglebutton
    ► trend
    ► trendfigure
  main-built.js
  main.js
  settings.js
```

datepicker

accordion

static map w link

client side trend

card layout area

link to a page parent

datepicker controlled content

toggle button

trend image

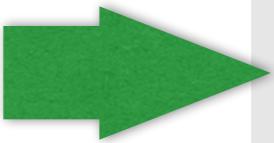
trend image in figure

css, images and static content

```
► app  
► build  
▼ css  
  ► favicons  
    accordion.css  
    alarmstatistics.content.css  
    app-bootstrap.css  
    app-cards.css  
    app.css  
    card.collection.css  
    card.event.css  
    card.pod.css  
    card.value.css  
    cason.png  
    collection.eventlimits.content.css  
    datepicker.css  
    event.page.css  
    ie10mobile.css  
    login.dialog.css  
    packeryarea.css  
    parentlink.css  
    report.page.css  
    series.content.css  
    shell.css  
    system.page.css  
  
► lib  
  index-build.html  
  index.html  
  License.txt  
  sc6.sublime-project  
  sc6.sublime-workspace  
  web.config  
  weyland-config.js
```

drop folder for the build process. not important in dev time.

```
► app  
► build  
▼ css  
  ► favicons  
    accordion.css  
    alarmstatistics.content.css  
    app-bootstrap.css  
    app-cards.css  
    app.css  
    card.collection.css  
    card.event.css  
    card.pod.css  
    card.value.css  
    cason.png  
    collection.eventlimits.content.css  
    datepicker.css  
    event.page.css  
    ie10mobile.css  
    login.dialog.css  
    packeryarea.css  
    parentlink.css  
    report.page.css  
    series.content.css  
    shell.css  
    system.page.css  
  
► lib  
  index-build.html  
  index.html  
  License.txt  
  sc6.sublime-project  
  sc6.sublime-workspace  
  web.config  
  weyland-config.js
```



```
► app  
► build  
► css  
▼ lib  
  ► bootstrap  
  ► bootstrap-datepicker  
  ► bootstrap-datepicker.old  
  ► bootstrap.old  
  ► d3  
  ► durandal  
  ► font-awesome  
  ► isotope  
  ► jquery  
  ► jquery-detectswipe  
  ► knockout  
  ► modernizr  
  ► moment  
  ► packery  
  ► require  
  ► spin  
  ► spin  
  ► unorm  
  index-build.html  
  index.html  
  License.txt  
  sc6.sublime-project  
  sc6.sublime-workspace  
  web.config  
  weyland-config.js
```

ui library (styles, components)

datepicker component

application framework

icon font

general use js library

gesture detection library

binding library

datetime library

card layout library (isotope)

module loader

unicode normalizer library: handles special characters
when sorting foreign names

How does it work?

Startup

Example 1

- Startup in detail
- Module loader
- Shell
- Routing
- Composition
- Module anatomy
- Knockout bindings
- Widget anatomy

localhost/sc6/3/

DIWICON Silver Cloud

CASON

DIWICON Silver | Loading.. | DIW | Warnings | DIW | Termoplín Vara | Command Ackn | Systems | DIWIC | Components - E | Szakács Szilárd | Diacritic - Wikipedia | nadibogi's stream | Play is the Way | Twitter / CSM | 10 User Experience

Elements Network Sources Timeline Profiles Resources Audits Console KnockoutJS PageSpeed

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body screen_capture_injected="true">
    <script>...</script>
    <div class="container">...</div>
    <!-- DEVELOPMENT: -->
    <script src="lib/require/require.js" data-main="app/main"></script>
    <!-- PRODUCTION: -->
    <script src="app/main-built.js" data-main="app/main"></script>
```

html body

Styles Event Listeners DOM Breakpoints Properties Knockout context

```
element.style { }
@media (max-width: 700px)
body {
  padding: 0;
}
@media (max-width: 700px)
body {
  background-color: white;
}
body {
  background-color: #EDEEFF;
}
body {
  min-height: 100%;
  padding: 0;
  padding-top: 30px;
  padding-bottom: 30px;
}
```

app.css:50
app.css:58
app.css:32
app.css:33
app.css:28
app.css:21
app.css:17

margin - border - padding - auto x auto - - -

Show inherited properties

-webkit-animation-delay: 0s;
-webkit-animation-direction: normal;
-webkit-animation-duration: 0s;
-webkit-animation-fill-mode: none;
-webkit-animation-iteration-count: 1;

Console Search Emulation Rendering

Device Emulate screen

Screen

Resolution: 320 578

User Agent

Sensors

Device pixel ratio: 1

Enable text autosizing

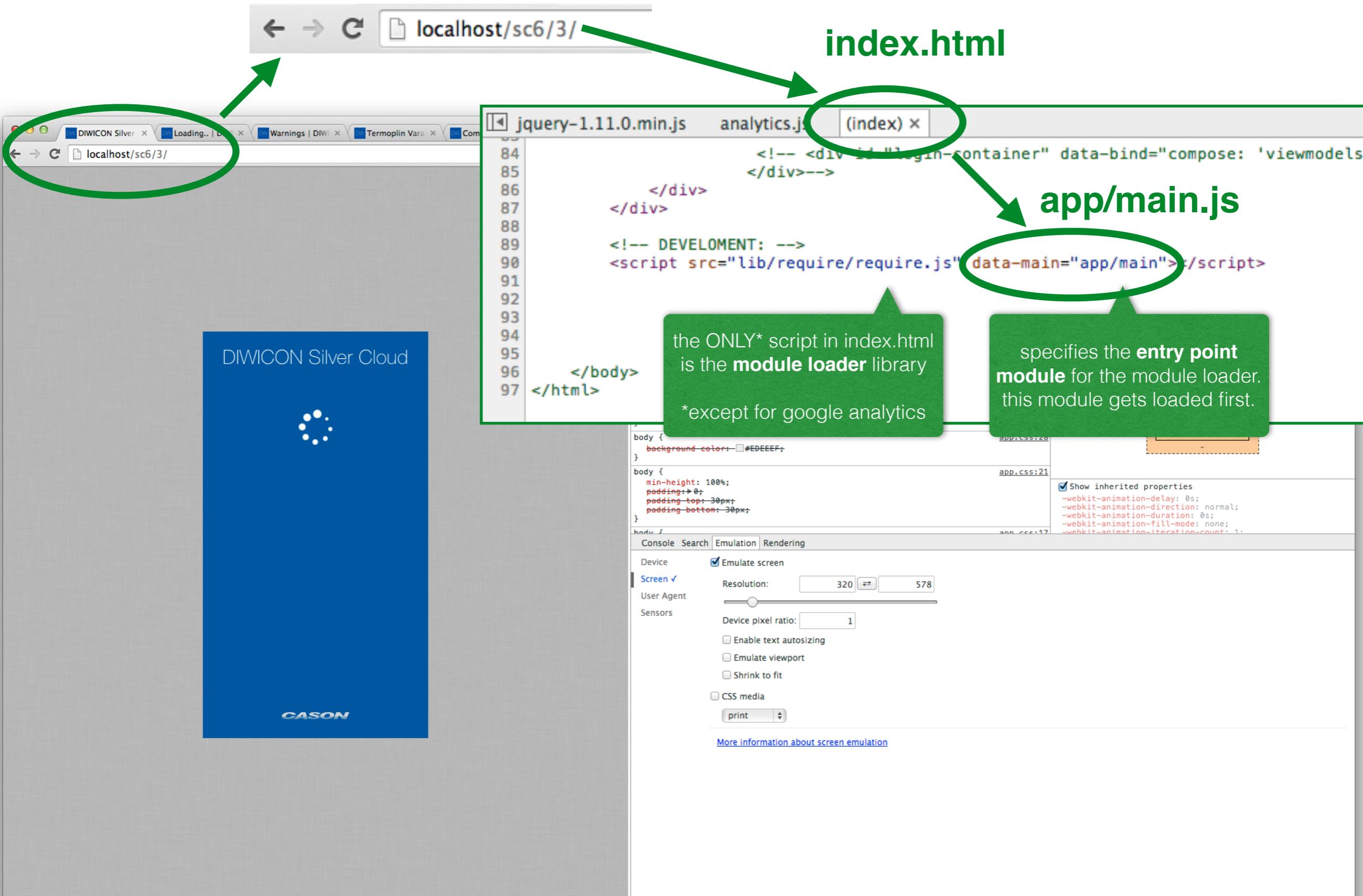
Emulate viewport

Shrink to fit

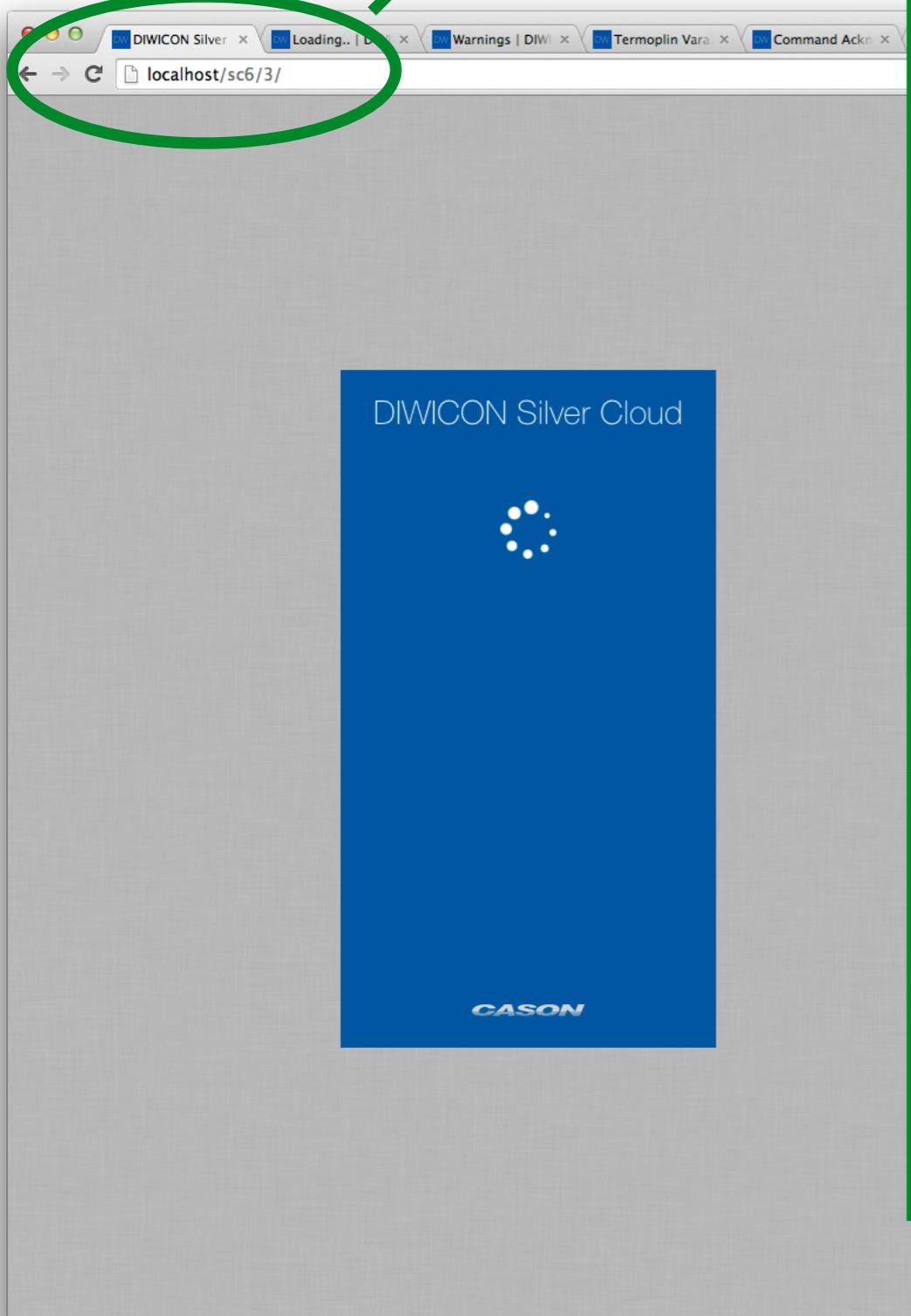
CSS media

print

[More information about screen emulation](#)



index.html → app/main.js



```
main.js x
1 requirejs.config({
2     waitSeconds: 120, // timeout for async require.js plugin: https://github.com/millermendeiros/re
3     paths: {
4         'async': '../lib/require/async',
5         'text': '../lib/require/text',
6         'durandal': '../lib/durandal/js',
7         'plugins' : '../lib/durandal/js/plugins',
8         'transitions' : '../lib/durandal/js/transitions',
9         'knockout': '../lib/knockout/knockout-2.3.0.debug',
10        'bootstrap': '../lib/bootstrap/js/bootstrap',
11        'jquery': '../lib/jquery/jquery-1.9.1',
12        'moment': '../lib/moment/moment-with-langs.min',
13        'spin': '../lib/spin/spin.min',
14        'datepicker': '../lib/bootstrap-datepicker/bootstrap-datepicker',
15        'unorm': '../lib/unorm/unorm',
16        'unormdata': '../lib/unorm/unormdata',
17        'detectswipe': '../lib/jquery-detectswipe/jquery.detect_swipe',
18        'packery': '../lib/packery/packery.pkgd'
19    },
20    shim: {
21        'bootstrap': {
22            deps: ['jquery'],
23            exports: 'jQuery'
24        },
25        'datepicker': {
26            deps: ['jquery'],
27            exports: 'jQuery.fn.datepicker'
28            //TODO $('[data-datepicker]').datepicker(); in custom binding
29        },
30        'unorm': {
31            exports: 'unorm'
32        },
33        'unormdata': {
34            deps: ['unorm'],
35            exports: 'unormdata'
36        },
37        'detectswipe': {
38            deps: ['jquery'],
39            exports: 'detectswipe'
40        }
41    }
42});
43
44 requirejs.onError = function (err) {
45     console.log(err.requireType);
46     if (err.requireType === 'timeout') {
47         console.log('modules: ' + err.requireModules);
48     }
49
50     throw err;
51};
52
53
54
55
```

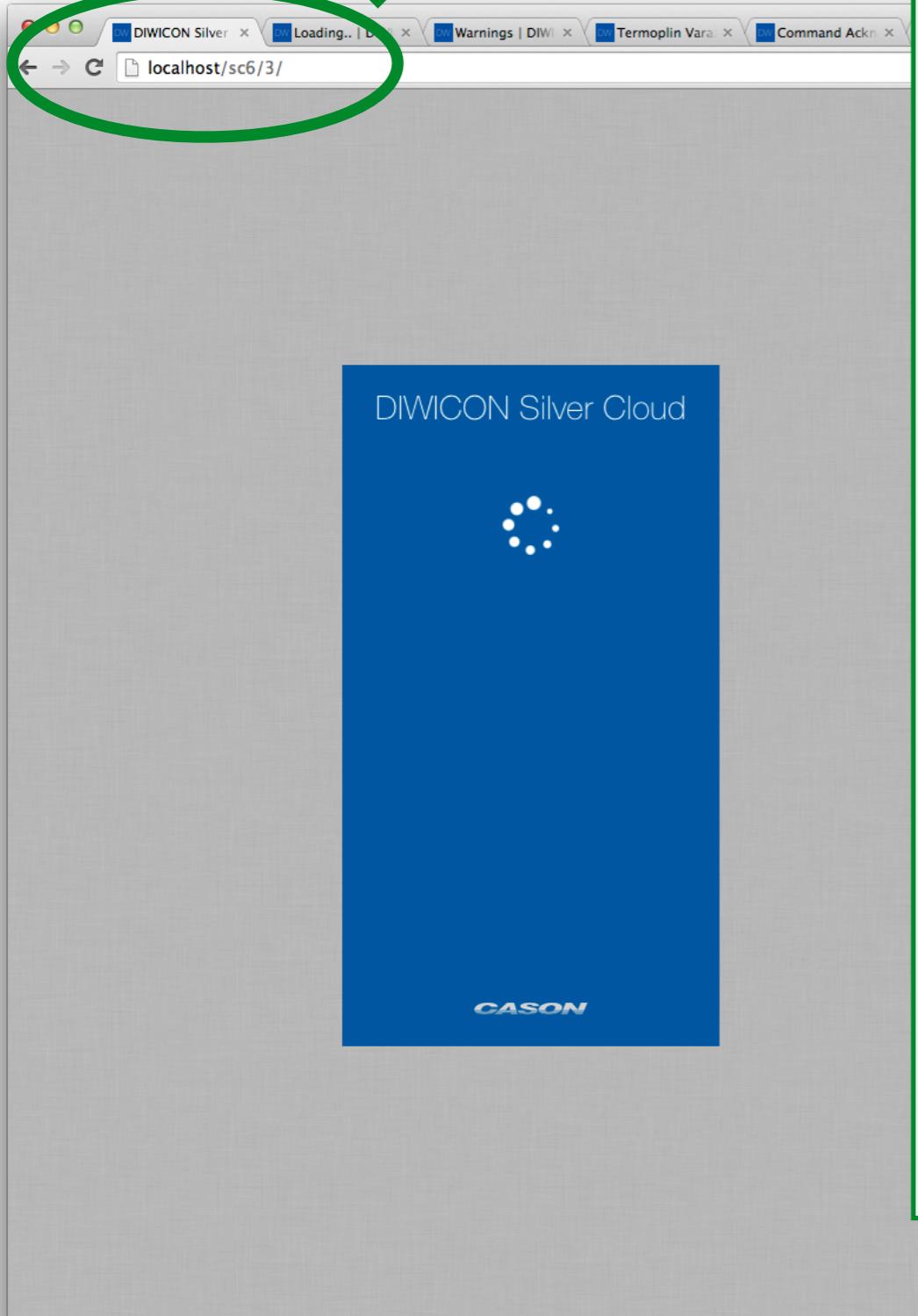
main.js 1/3
module loader
configuration

index.html → app/main.js

A screenshot of a web browser window. The address bar at the top shows the URL "localhost/sc6/3/" with a green oval and arrow highlighting it. The main content area displays a blue page titled "DIWICON Silver Cloud" with a circular loading icon in the center. The word "CASON" is visible at the bottom of the page. The browser has multiple tabs open in the background.

main.js 2/3

index.html → app/main.js



```
main.js x
89     try{
90         if(window.applicationCache){
91             window.applicationCache.update();
92             window.applicationCache.swapCache();
93         }
94     }catch(ex){}
95
96
97
98
99     moment.lang(settings.language ? settings.language : 'en');
100
101    app.title = 'DIWICON Silver Cloud';
102
103    var initShell = function(){
104        //Show the app by setting the root
105        app.setRoot('contexts/client/module');
106    };
107
108    app.configurePlugins({
109        router:true,
110        dialog: true,
111        widget: true
112    });
113
114    app.start().then(function() {
115        //Replace 'viewmodels' in the moduleId with 'views' to locate the view.
116        //Look for partial views in a 'views' folder in the root.
117        viewLocator.useConvention();
118
119        var showLogin = function(){
120            //HACK
121            $('#splash-spinner').hide();
122
123            app.on('login:success').then(function(){
124                initShell();
125            });
126
127            var hostElement = document.getElementById('login-container');
128            var settings = { activate: true, transition: 'slide-left' };
129            settings.model = 'contexts/client/modules/login';
130            composition.compose(hostElement, settings);
131        };
132
133        translationService.initTranslations().always(function(){
134            //continue app init
135            if(authenticationService.isAuthenticated()){
136                authenticationService.getCurrentUser().then(
137                    function(){
138                        // authentication token is valid
139                        initShell();
140                    },
141                    function(){
142                        // authentication token is invalid
143                        showLogin();
144                    }
145                );
146                // no authentication token present
147                else{
148                    showLogin();
149                }
150            });
151        });
152    });

```

**application entry point
called by durandal
application framework**

1. init translations

2. check authentication

2.1 init shell

2.2 or show login

main.js 3/3

Screenshot of a web browser showing the DIWICON Silver Cloud login page and the developer tools Elements tab.

The browser window title is "localhost/sc6/3/". The developer tools are open, showing the DOM structure and CSS styles for the current page.

DOM Structure:

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body screen_capture_injected="true">
    <script>...</script>
    <div class="container">...</div>
    <!-- DEVELOPMENT: -->
    <script src="lib/require/require.js" data-main="app/main"></script>
    <!-- PRODUCTION: -->
    <script src="app/main-built.js" data-main="app/main"></script>
```

Styles (CSS):

```
element.style { }
@media (max-width: 700px)
body {
  padding: 0;
}
@media (max-width: 700px)
body {
  background-color: white;
}
body {
  background-color: #EDEEFF;
}
body {
  min-height: 100%;
  padding: 0;
  padding-top: 30px;
  padding-bottom: 30px;
}
```

Emulation Tab:

- Device: Screen (selected)
- Resolution: 320 x 578
- User Agent: (dropdown menu)
- Sensors: (checkboxes)
- Device pixel ratio: 1
- Enable text autosizing: (checkbox)
- Emulate viewport: (checkbox)
- Shrink to fit: (checkbox)
- CSS media: (checkbox)
- print (button)

[More information about screen emulation](#)

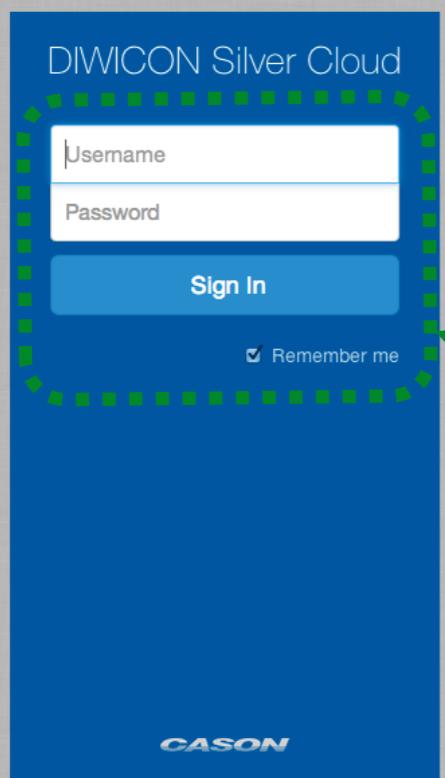
Right Panel:

- margin - border - padding - auto x auto
- Show inherited properties (checkbox checked)
- webkit-animation-delay: 0s;
-webkit-animation-direction: normal;
-webkit-animation-duration: 0s;
-webkit-animation-fill-mode: none;
-webkit-animation-iteration-count: 1;

index.html → app/main.js

The screenshot shows a browser window with the URL `localhost/sc6/3/` circled in green at the top left. Below it is a file explorer interface showing the project structure:

- app
 - contexts
 - api
 - client
 - bindings
 - handlers
 - models
 - modules
 - login
 - dialog.html
 - dialog.js



view +
viewmodel

The screenshot shows a code editor with the file `main.js` open. The code is written in JavaScript and handles the initialization of the application. A green callout bubble points to the section of code that composes the login dialog.

```
try{
    if(window.applicationCache){
        window.applicationCache.update();
        window.applicationCache.swapCache();
    }
} catch(ex){}

moment.lang(settings.language ? settings.language : 'en');

app.title = 'DIWICON Silver Cloud';

var initShell = function(){
    //Show the app by setting the root view model for our application with a transition.
    app.setRoot('contexts/client/modules/shell/shell', 'entrance');
};

app.configurePlugins({
    router:true,
    dialog: true,
    widget: true
});

app.start().then(function() {
    //Replace 'viewmodels' in the moduleId with 'views' to locate the view.
    //Look for partial views in a 'views' folder in the root.
    viewLocator.useConvention();

    var showLogin = function(){
        //HACK
        $('#splash-spinner').hide();

        app.on('login:success').then(function(){
            initShell();
        });

        var hostElement = document.getElementById('login-container');
        var settings = { activate: true, transition: 'entrance' };
        settings.model = 'contexts/client/modules/login/dialog';
        composition.compose(hostElement, settings);
    };

    translationService.initTranslations().always(function(){
        //continue app init
        if(authenticationService.isAuthenticated()){
            authenticationService.getCurrentUser().then(
                function(){
                    // authentication token is validated by server
                    initShell();
                },
                function(){
                    // authentication token is invalid
                    showLogin();
                }
            );
        }
        // no authentication token present
        else{
            showLogin();
        }
    });
});
});
```

A green callout bubble on the right side of the code editor points to the `composition.compose` line and contains the text "compose login dialog". Another green callout bubble at the bottom right points to the line number 152 and contains the text "main.js 3/3".

view +.viewmodel

login/dialog.html

```

<section id="login-dialog">
    <!--<div class="container">
        <div class="row">
            <div class="col-sm-6 col-md-4 col-md-offset-4">-->

                <div class="account-wall">
                    <!--<h1 class="text-center login-title">Log in</h1>-->
                    <form class="form-signin" data-bind="submit: login">
                        <div class="alert alert-danger" data-bind="show: hasError, text: errorMessage">
                            <input id="login-username" type="text" class="form-control" data-bind="value: username, valueUpdate: 'afterkeydown', disabled: isDisabled, translate: {placeholder: 'Login_Username_Placeholder'}" required="" autofocus>
                            <input type="password" data-bind="value: password, valueUpdate: 'afterkeydown', disabled: isDisabled, translate: {placeholder: 'Login_Password_Placeholder'}" required="" placeholder="Password" type="password"/>
                        <button id="login-button" class="btn btn-lg btn-primary btn-block has-spinner" type="submit" data-bind="disabled: isDisabled, css: {active: isLoading, 'btn-primary': !isSuccess(), 'btn-success': isSuccess}">
                            <span class="spinner"><i class="icon icon-spin icon-white" data-bind="text: spinnerText" /></span>
                            <!-- ko text: loginButtonLabel -->
                        </button>
                        <label id="login-remember-me" class="checkbox pull-right">
                            <input type="checkbox" value="remember-me" data-bind="disabled: isDisabled, checked: rememberMe, translate: {text: 'Login_Remember_Me'}" checked="checked"/>
                            Remember me
                        </label>
                    <!-- TODO <a href="#" data-bind="click: forgotPassword">Forgot password? </a> -->
                    <span class="clearfix">
                    </span>
                </form>
            </div>
            <!--<a href="#" class="text-center new-account">Create an account </a>-->
        <!--</div>
    </div>
</div>-->
</section>

```

a function is bound to the submit action of the form

translating the placeholder property of the element

btn, btn-lg, ... - **Bootstrap.js** ui library's built in classes. these provide consistent styling to the ui.

data-bind: **knockout.js** declarative binding to viewmodel variables

translate binding: the best way to localize the ui. translating the text property of the element

DIWICON Silver Cloud

localhost/sc6/3/

```

app
  contexts
    api
  client
    bindings
    handlers
    models
  modules
    login
      dialog.html
      dialog.js

```

view +.viewmodel

```

2 ▼ define([
3   'settings',
4   'knockout',
5   'durandal/system',
6   'durandal/app',
7   'plugins/router',
8   'contexts/api/services/urlMapper',
9   'contexts/api/services/translation',
10  'contexts/api/services/authentication',
11  'contexts/client/services/pageHelper'
12 ],
13 ▼ function (
14   settings,
15   ko,
16   system,
17   app,
18   router,
19   apiUrlMapper,
20   translationService,
21   authenticationService,
22   pageHelpers) {
23
24   return {
25     usernameOrEmail: ko.observable(),
26     password: ko.observable(),
27     errorMessage: ko.observable(null),
28     isLoading: ko.observable(false),
29     isEnabled: ko.observable(false),
30     isSuccess: ko.observable(false),
31     rememberMe: ko.observable(true),
32     login: function(){
33       this.errorMessage(null);
34       var self = this;
35       this.isLoading(true);
36       this.isEnabled(true);
37
38       return authenticationService.login(this.usernameOrEmail(), this.password(), this.rememberMe())
39         // translate the message
40         .always(function(loginResult){
41           return system.defer(function(dfdf){
42             translationService.translate(loginResult.message).then(function(translation){
43               loginResult.message = translation;
44               dfdf.resolve(loginResult);
45             },
46             // resolve even if the translation does not resolve
47             function(){
48               dfdf.resolve(loginResult);
49             });
50           }).promise();
51         }
52         // update view state
53         .always(function(loginResult){
54           if(loginResult.isSuccess){
55             self.isSuccess(true);
56             translationService.translate('Login_Button');
57               self.loginButtonLabel(translation);
58             });
59         })
59

```

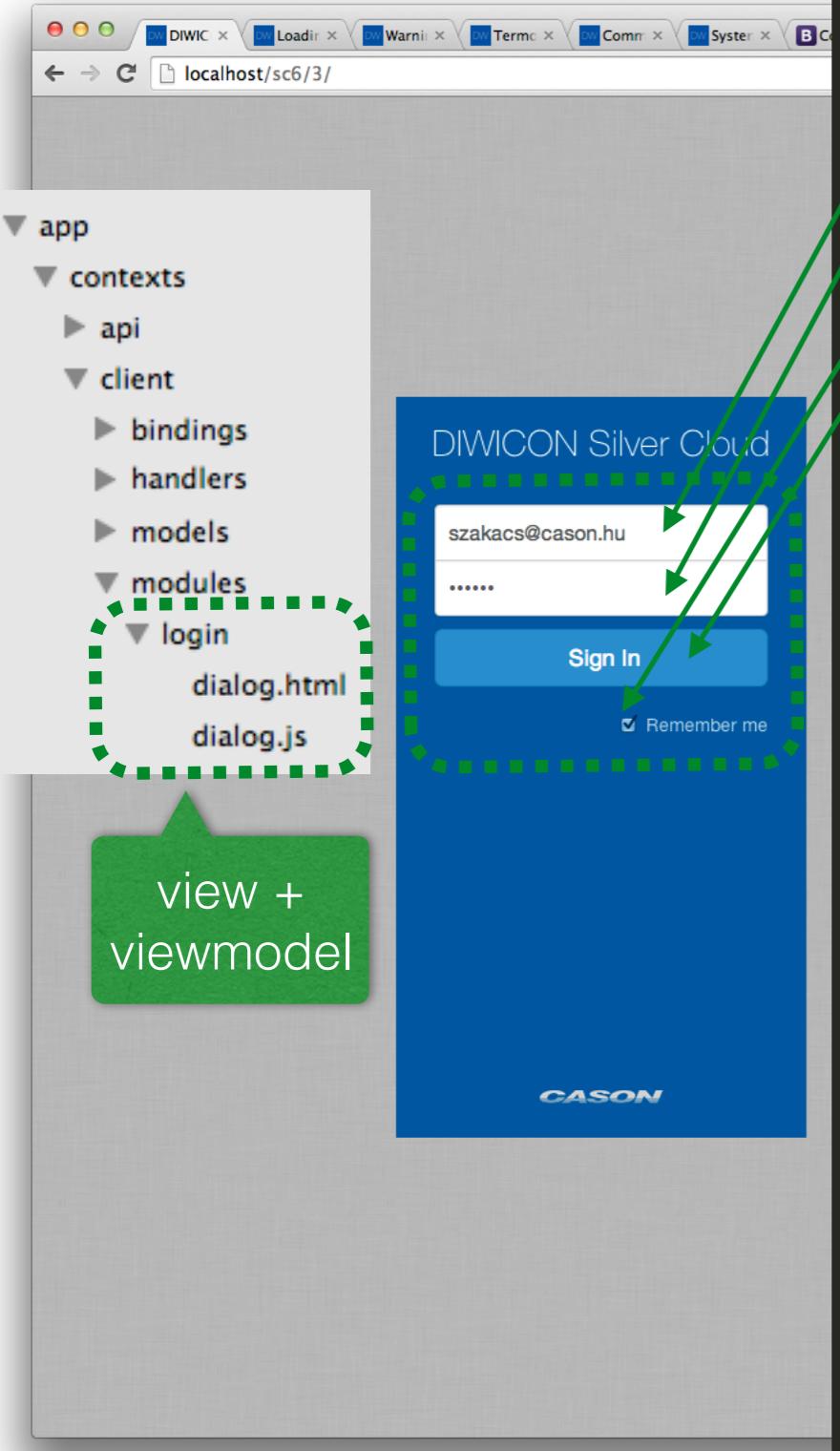
required modules loaded async and the function w the module params is only called of all of them (and all of their module dependencies) are called (see [Require.js Definition Functions](#))

function returns prototype-level public functions and knockout observable variables.

(see [Revealing Module Pattern](#))

(see [Knockout Observables](#))

login/dialog.js 1/2



```

24     return {
25       usernameOrEmail: ko.observable(),
26       password: ko.observable(),
27       errorMessage: ko.observable(null),
28       isLoading: ko.observable(false),
29       isDisabled: ko.observable(false),
30       isSuccess: ko.observable(false),
31       rememberMe: ko.observable(true),
32       login: function(){
33         this.errorMessage(null);
34         var self = this;
35         this.isLoading(true);
36         this.isDisabled(true);
37
38         return authenticationService.login(this.usernameOrEmail(), this.password(), this.rememberMe())
39           // translate the message
40           .always(function(loginResult){
41             return system.defer(function(dfdf){
42               translationService.translate(loginResult.message).then(function(translation){
43                 loginResult.message = translation;
44                 dfdf.resolve(loginResult);
45               },
46               // resolve even if the translation does not resolve
47               function(){
48                 dfdf.resolve(loginResult);
49               });
50             }).promise();
51           }
52           // update view state
53           .always(function(loginResult){
54             if(loginResult.isSuccess){
55               self.isSuccess(true);
56               translationService.translate('Login_Button_Success','Success!').then(function(translation){
57                 self.loginButtonLabel(translation);
58               });
59             }
56             else{
57               self.errorMessage(loginResult.message);
58               self.isDisabled(false);
59               self.isSuccess(false);
60             }
61           }
62           .always(function(){
63             self.isLoading(false);
64           });
65           // life-cycle hooks
66           activate: function () {
67             this.loginButtonLabel = ko.observable();
68             //TODELETE this.rememberMe(authenticationService.isRemembered());
69             return translationService.translate('Login_Button_SignIn','Sign in').then(function(translation){
70               this.loginButtonLabel(translation);
71             }).bind(this);
72           },
73           compositionComplete: function(view){
74             document.getElementById("login-username").focus();
75           }
76         );
77       });
78     };
79   });
80   });
81 });
82 });

```

observable variables that we bind
the ui against to

login/dialog.js 2/2

view +.viewmodel

The screenshot shows a login dialog box with the title 'DIWICON Silver Cloud'. It contains input fields for 'szakacs@cason.hu' and '.....', and a 'Sign In' button. A 'Remember me' checkbox is also present.

```

24   return {
25     usernameOrEmail: ko.observable(),
26     password: ko.observable(),
27     errorMessage: ko.observable(null),
28     isLoading: ko.observable(false),
29     isEnabled: ko.observable(false),
30     isSuccess: ko.observable(false),
31     rememberMe: ko.observable(true),
32     login: function(){
33       this.errorMessage(null);
34       var self = this;
35       this.isLoading(true);
36       this.isEnabled(true);
37
38       return authenticationService.login(this.usernameOrEmail(), this.password(), this.rememberMe())
39         // translate the message
40         .always(function(loginResult){
41           return system.defer(function(dfd){
42             translationService.translate(loginResult);
43             loginResult.message = translation;
44             dfd.resolve(loginResult);
45           },
46           // resolve even if the translation does
47           function(){
48             dfd.resolve(loginResult);
49           });
50         }).promise();
51
52       // update view state
53       .always(function(loginResult){
54         if(loginResult.isSuccess){
55           self.isSuccess(true);
56           translationService.translate('Login_Button_Success', 'Success!').then(function(translation){
57             self.loginButtonLabel(translation);
58           });
59         }
56       else{
57         self.errorMessage(loginResult.message);
58         self.isEnabled(false);
59         self.isSuccess(false);
60       }
61     })
62     .always(function(){
63       self.isLoading(false);
64     });
65
66     // life-cycle hooks
67     activate: function () {
68       this.loginButtonLabel = ko.observable();
69       //TODELETE this.rememberMe(authenticationService.isRemembered());
70       return translationService.translate('Login_Button_SignIn', 'Sign in').then(function(translation){
71         this.loginButtonLabel(translation);
72         }.bind(this));
73     },
74     compositionComplete: function(view){
75       document.getElementById("login-username").focus();
76     }
77   );
78
79   });
80
81 });
82 });

```

calls login on the required authentication application service module **ASYNCHRONOUSLY** (see [jQuery Promises](#))

you can resolve or reject a deferred marking the async operation successful or failed.

returns a promise because the translation in the function is async. it uses durandal helper module [durandal/system](#). This is the way you can create your own promises.

login/dialog.js 2/2

login/dialog.js

```
login: function(){
```

```
    this.errorMessage(null);  
    var self = this;  
    this.isLoading(true);  
    this.isDisabled(true);
```

```
    return authenticationService.login(this.usernameOrEmail(), this.password(), this.rememberMe())  
        .always(function(loginResult){  
            return system.defer(function(dfdf){  
                translationService.translate(loginResult.message).then(function(translation){  
                    loginResult.message = translation;  
                    dfd.resolve(loginResult);  
                },  
                // resolve even if the translation does not resolve  
                function(){  
                    dfd.resolve(loginResult);  
                });  
            }).promise();  
        })  
        .always(function(result){  
            if(result.error){  
                log.error(result.error);  
            }  
        })
```

DIWICON Silver Cloud

szakacs@cason.hu

.....

Sign In

Remember me

CASON

```
▼ app  
  ▼ contexts  
    ▶ api  
    ▼ client  
      ▶ bindings  
      ▶ handlers  
      ▶ models  
    ▼ modules  
      ▼ login  
        dialog.html  
        dialog.js
```

contexts/api/services/authentication.js

```
var login = function(usernameOrEmail, password, rememberMe){
```

```
    return $.post(settings.apiRoot + 'users/authenticate', {id: usernameOrEmail, password: password})  
        .always(function(result){
```

```
            app.trigger('login:success', null);
```

```
            var userToken = result.token;
```

```
            if(rememberMe){
```

```
                setRemeberedCredentials(userToken);
```

```
            } else {
```

```
                clearRememberedCredentials();
```

```
            }  
        },  
        // on error  
        function(httpObj, textStatus){  
            var failMessage = 'Login Error';
```

```
            app.trigger('login:error', failMessage);
```

```
        }  
    );  
};
```

Successful login event is advertised on the application level pubsub **messaging bus**.

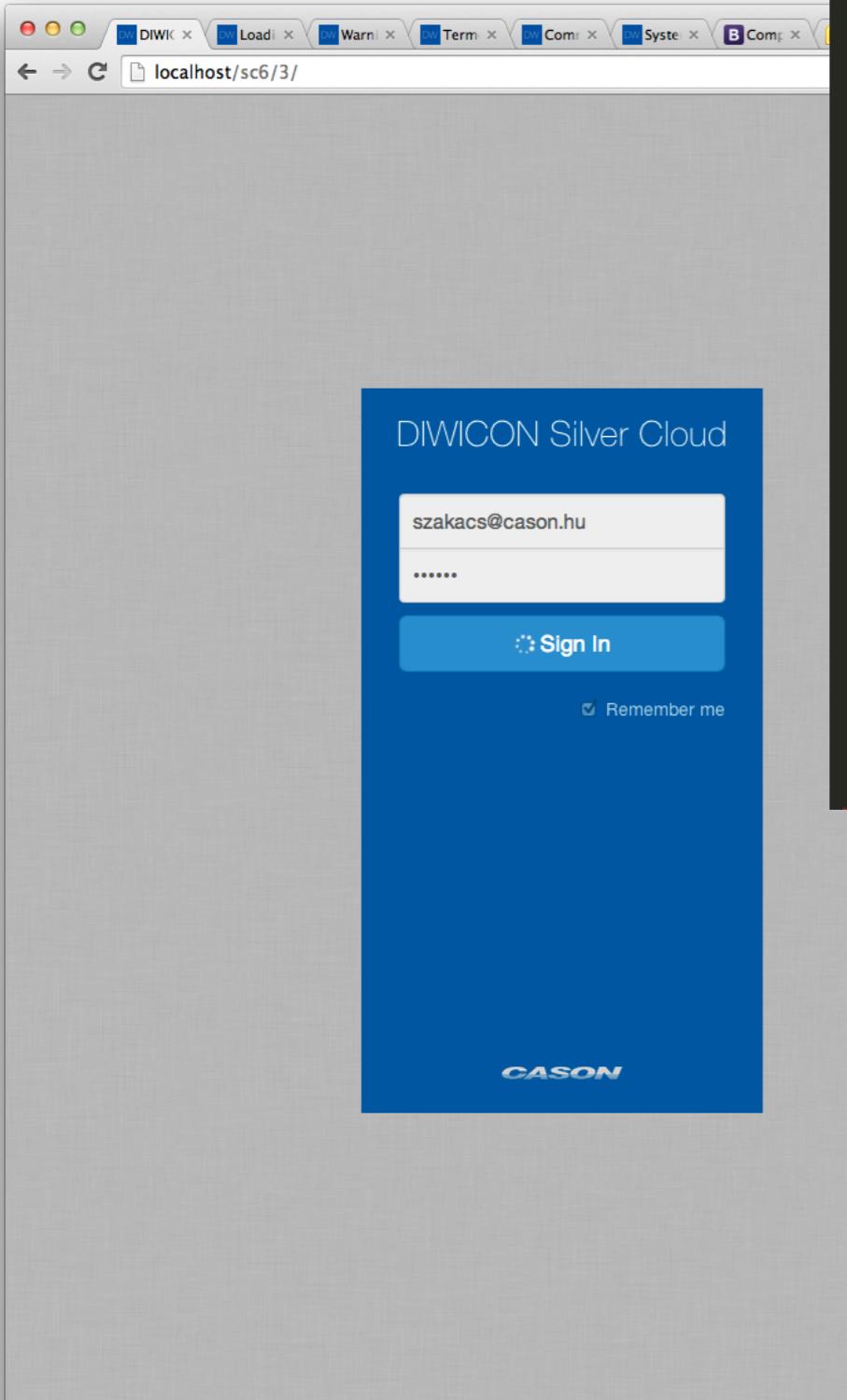
This is a lightweight way to exchange data between modules.

(See [Durandal Events](#))

app/main.js

```
app.start().then(function() {  
    //Replace 'viewmodels' in the moduleId with 'views'  
    //Look for partial views in a 'views' folder in viewLocation  
    viewLocator.useConvention();  
  
    var showLogin = function(){  
        //HACK  
        $('#splash-spinner').hide();  
  
        app.on('login:success').then(function(){  
            initShell();  
        });  
  
        var hostElement = document.getElementById('login');  
        var settings = { activate: true, transition: 'slide-left' };  
        settings.model = 'contexts/client/modules/login';  
        composition.compose(hostElement, settings);  
    };  
});
```

The main module (who initiated the login process) learns about the successful login and **initiates the shell** module (starts the application basically).



the shell's viewmodel gets **activated** and
after the successful activation the content of the applicationHost div gets exchanged with the shell's view bound to its viewmodel.

(See [Durandal's Module Lifecycle callbacks](#))

```

</script>
<div class="container">
  <div id="applicationHost">
    ...
  </div>
</div>

<!-- DEVELOPMENT: -->
<script src="lib/require/require.js" data-main="app/main"></script>

<!-- PRODUCTION:
<script src="app/main-built.js" data-main="app
-->

</body>

```

index.html

```

100 app.title = 'DIWICON Silver Cloud';
101
102 var initShell = function(){
103   //Show the app by setting the root view model for our application with a transition.
104   app.setRoot('contexts/client/modules/shell/shell', 'entrance');
105 };
106
107 app.configurePlugins({
108   router: true,
109   dialog: true,
110   widget: true
111 });
112
113 app.start().then(function() {
114   //Replace 'viewmodels' in the moduleId with 'views'
115   //Look for partial views in a 'views' folder in the module
116   viewLocator.setConvention();
117
118   var showLogin = function(){
119     //HACK
120     $('#splash-spinner').hide();
121
122     app.on('login:success').then(function(){
123       initShell();
124     });
125
126     var hostElement = document.getElementById('login');
127     var settings = { activate: true, transition: 'slide-left' };
128     settings.model = 'contexts/client/modules/login';
129     composition.compose(hostElement, settings);
130   };
131
132 });

```

app/main.js

The main module (who initiated the login process) learns about the successful login and **initiates the shell** module (starts the application basically).

localhost/sc6/3/#

Connection Error Reload page

Home

Elements Network Sources Timeline Profiles Resources Audits Console KnockoutJS PageSpeed

Sources Content scripts Snippets main.js

```
// force iOS app cache update
try{
    if(window.applicationCache){
        window.applicationCache.update();
        window.applicationCache.swapCache();
    }
} catch(ex){}

moment.lang(settings.language ? settings.language : 'en');

app.title = 'DIWICON Silver Cloud';

var initShell = function(){
    //Show the app by setting the root view model for our application with a transition.
    app.setRoot('contexts/client/modules/shell/shell', 'entrance');
};

app.configurePlugins({
    router:true,
    dialog: true,
    widget: true
});

app.start().then(function() {
    //Replace 'viewmodels' in the moduleId with 'views' to locate the view.
    //Look for partial views in a 'views' folder in the root.
    viewLocator.useConvention();

    var showLogin = function(){
        //HACK
        $('#splash-spinner').hide();
        app.on('login:success').then(function(){
            initShell();
        });

        var hostElement = document.getElementById('login-container');
        var settings = { activate: true, transition: 'entrance' };
        settings.model = 'contexts/client/modules/login/dialog';
        composition.compose(hostElement, settings);
    };

    translationService.initTranslations().always(function(){
        //continue app init
        if(authenticationService.isAuthenticated()){
            authenticationService.getCurrentUser().then(
                function(){
                    // authentication token is validated by server
                    initShell();
                },
                function(){
                    // authentication token is invalid
                    showLogin();
                }
            );
        }
        // no authentication token present
        else{
            showLogin();
        }
    });
});
});
```

{} Line 106, Column 7

Scope Variables Watch Expressions

Console Search Emulation Rendering

Device Emulate screen

Screen Resolution: 770 578

navigation host.
router loads
modules here.

```

1   <div>
2
3     <div id="applicationErrorHost"
4       style="display:block; padding: .5em; margin-top: .5em; background-color: WhiteSmoke ; color:
5       gray; border-radius: 4px"
6       data-bind="fadeVisible: dataService.isTimeout">
7       <span class="glyphicon glyphicon-flash"></span> <span data-bind="translate: {text: 'Connection
8       Error'}" >Connection error</span> <a href="#" data-bind="click: dataService.retryTimeout,
9       translate: {text: 'Reload page'}>Reload page</a>
10      </div>
11
12      <!--ko widget: {kind: 'parentlink'}--><!--/ko-->
13
14      <div class="container-fluid page-host" data-bind="route">
15        </div>
16
17      <div class="page-footer">
18        <ul class="list-inline" data-bind="fadeVisible: loginVisible">
19          <li>
20            <span class="glyphicon glyphicon-user" style="font-size: 1.5em; font-weight: bold" data-bind="visible: visible"></span>
21            <span style="font-weight: bold" data-bind="visible: visible">User</span>
22          </li>
23          <!-- ko foreach: menuList -->
24          <li data-bind="visible: visible">
25            <a data-bind="attr: { href: href }, translate: {text: 'Logout'}>Logout</a>
26          </li>
27        </ul>
28      </div>

```

shell.html

Warning! This is not a deletable comment!

It's a Knockout.js virtual element binding, meaning it does not need a HTML element as a host. This is beneficial for performance reasons.

szakacs@cason.hu Settings Log out

localhost/sc6/3/#

app contexts api client bindings handlers models modules login shell shell.html shell.js

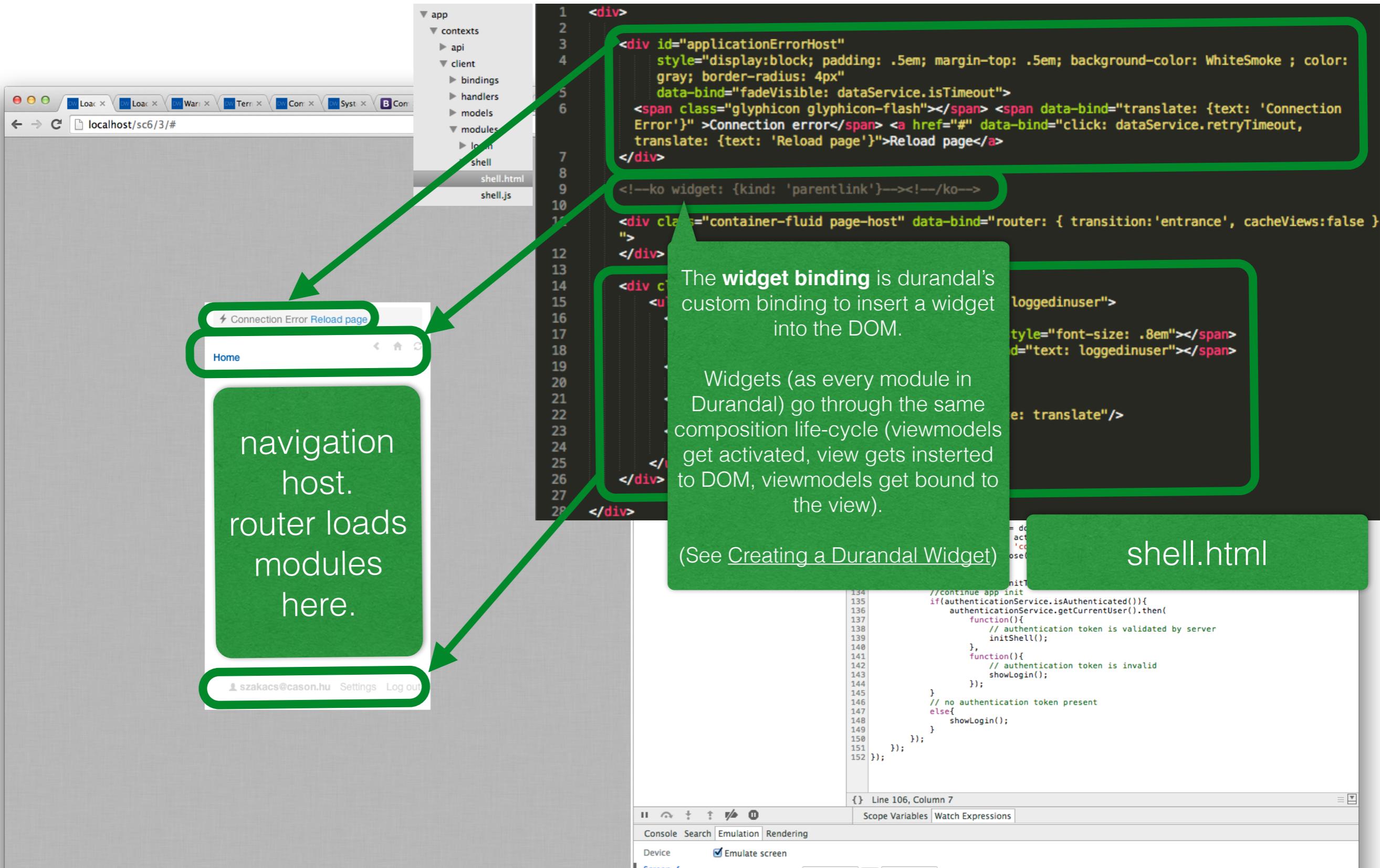
127 var hostElement = document.getElementById('applicationErrorHost');
128 var settings = { activeTab: 'home', settingsModel: {} };
129 settings.model = 'connectionError';
130 composition.compose(settings);
131
132
133 translationService.init();
134 //continue app init
135 if(authenticationService.isAuthenticated()){
136 authenticationService.getCurrentUser().then(
137 function(){
138 // authentication token is validated by server
139 initShell();
140 },
141 function(){
142 // authentication token is invalid
143 showLogin();
144 }
145);
146 // no authentication token present
147 else{
148 showLogin();
149 }
150 }
151 });
152);

{} Line 106, Column 7

Console Search Emulation Rendering

Device Emulate screen

Screen Resolution: 720x578



```

return {
  router: router,
  activate: function () {

    this.dataService = dataService;
    this.loggedinuser = ko.observable();
    this.menuList = ko.observableArray([{ href: '#/logout', text: 'Log out', translate: { text: 'Log_out' }, visible: 1 }]);

    authenticationService.getCurrentUser().then(function(user){
      this.loggedinuser(user.Id);
      this.menuList([
        { href: '#/management', text: 'Management', translate: { text: 'Management' }, visible: user.Roles.indexOf("Superuser") > -1 },
        { href: '#/users/current/settings', text: 'Settings', translate: { text: 'Settings' }, visible: 1 },
        { href: '#/logout', text: 'Log out', translate: { text: 'Log_out' }, visible: 1 }
      ]);
    }.bind(this));

    // CONVENTIONS
    // * For pages using
    // * Only one route per module
    // * The startup route

    // TODO same as the viewmodel service. move to common strategy module.
    var getResourcePage = function(resourceType, isEdit){
      return apiLocator.getPageId(resourceType, null, isEdit);
    };

    apiUrlMapper.setClientHomeApi(settings.apiRoot + 'systems');

    router.map([
      { route: '', title: 'Home', moduleId: getResourcePage('collection') },
      { route: 'login', title: 'Login', moduleId: 'viewmodels/login.page' },
      { route: 'management', title: 'Management', moduleId: getResourcePage('user', true) },
      { route: 'users/current/settings', title: 'Settings', moduleId: 'contexts/api/modules/profile/page' },
      { route: 'logout', title: 'Logging out...', moduleId: 'contexts/client/handlers/logout' },

      { route: 'systems', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId', title: 'Loading...', moduleId: getResourcePage('system') },
      { route: 'systems/:systemId/events/:eventGroup/:eventCollection', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/events/:eventGroup/:eventCollection/:eventId', title: 'Loading...', moduleId: getResourcePage('event') },

      { route: 'systems/:systemId/pods', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/pods/:podId', title: 'Loading...', moduleId: getResourcePage('pod') },

      { route: 'systems/:systemId/pods/:podId/events', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/pods/:podId/events/:eventGroup/:eventCollection', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/pods/:podId/events/:eventGroup/:eventCollection/:eventId', title: 'Loading...', moduleId: getResourcePage('event') },

      { route: 'systems/:systemId/pods/:podId/measurements', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/pods/:podId/measurements/:channelType/quantities', title: 'Loading...', moduleId: getResourcePage('collection') },
      { route: 'systems/:systemId/pods/:podId/measurements/:channelType/quantities/:quantity(/:date)', title: 'Loading...', moduleId: getResourcePage('series') }
    ]).buildNavigationMenu();

    router.guardRoute = function(routeInfo, params, instance) {
      var result = authenticationService.isAuthenticated();
      if(!result) {
        result = 'login';
      }
      return result;
    };

    router.mapUnknownRoutes();

    return router.activate();
  }
};

```

client route table. Maps hashes to module ids. The matching route instruction will load the module identified by its module id to the shell's navigation host.

(See Durandal Router Plugin)

This gets executed before every navigation.

Activates the router, which inturn activates the route instruction matching the current hash.

The api context's urlMapper app service can convert every client path to api path. Most pages use this to get their data. Except for the home page, as its client path is empty. Therefore we need to provide this api address manually.

shell.js

Router checks the hash.

```

localhost/sc6/3 #
```

Router selects first matching route instruction and loads the module specified by the moduleId - here generated by convention from a resource's type.

```

this.loggedinuser = ko.observable();
this.menuList = ko.observableArray([{ href: '#/logout', text: 'Log out', translate: { text: 'Log_out' } }]);

authenticationService.getCurrentUser().then(function(user){
    this.loggedinuser(user.Id);
    this.menuList([
        { href: '#/management', text: 'Management', translate: { text: 'Management' }, visible: user.IsAdmin },
        { href: '#/users/current/settings', text: 'Settings', translate: { text: 'Settings' }, visible: true },
        { href: '#/logout', text: 'Log out', translate: { text: 'Log_out' }, visible: 1 }
    ]);
}).bind(this));

// CONVENTION BASED ROUTING
// * For each resource type there is a corresponding module
// * On each module there is a resourceType
// * The resourceType is mapped to the URL via the router
// * TODO: Implement a better way to handle this
var getResourcePage = function(resourceType) {
    return 'contexts/api/modules/' + resourceType + '/page';
};

angular.module('UrlMapper', [ 'ui.router' ])
.config(function($stateProvider, $urlRouterProvider) {
    $urlRouterProvider.otherwise('/');

    $stateProvider
        .state('home', {
            route: '',
            title: 'Home',
            moduleId: getResourcePage('collection')
        })
        .state('login', {
            route: '/login',
            title: 'Login',
            moduleId: 'views/modules/login/page'
        })
        .state('management', {
            route: '/management',
            title: 'Management',
            moduleId: getResourcePage('user', true)
        })
        .state('users/current/settings', {
            route: '/users/current/settings',
            title: 'Settings',
            moduleId: 'contexts/api/modules/profile/page'
        })
        .state('logout', {
            route: '/logout',
            title: 'Logging out..',
            moduleId: 'contexts/client/handlers/logout'
        })

        .state('systems', {
            route: '/systems',
            title: 'Loading..',
            moduleId: getResourcePage('collection')
        })
        .state('systems/:systemId', {
            route: '/systems/:systemId',
            title: 'Loading..',
            moduleId: getResourcePage('system')
        })
        .state('systems/:systemId/events/:eventGroup/:eventCollection', {
            route: '/systems/:systemId/events/:eventGroup/:eventCollection',
            title: 'Loading..',
            moduleId: getResourcePage('eventCollection')
        })
        .state('systems/:systemId/events/:eventGroup/:eventId', {
            route: '/systems/:systemId/events/:eventGroup/:eventId',
            title: 'Loading..',
            moduleId: getResourcePage('event')
        })

        .state('systems/:systemId/pods', {
            route: '/systems/:systemId/pods',
            title: 'Loading..',
            moduleId: getResourcePage('collection')
        })
        .state('systems/:systemId/pods/:podId', {
            route: '/systems/:systemId/pods/:podId',
            title: 'Loading..',
            moduleId: getResourcePage('pod')
        })

        .state('systems/:systemId/pods/:podId/events', {
            route: '/systems/:systemId/pods/:podId/events',
            title: 'Loading..',
            moduleId: getResourcePage('collection')
        })
        .state('systems/:systemId/pods/:podId/events/:eventGroup/:eventCollection', {
            route: '/systems/:systemId/pods/:podId/events/:eventGroup/:eventCollection',
            title: 'Loading..',
            moduleId: getResourcePage('eventCollection')
        })
        .state('systems/:systemId/pods/:podId/events/:eventGroup/:eventId', {
            route: '/systems/:systemId/pods/:podId/events/:eventGroup/:eventId',
            title: 'Loading..',
            moduleId: getResourcePage('event')
        })

        .state('systems/:systemId/pods/:podId/measurements', {
            route: '/systems/:systemId/pods/:podId/measurements',
            title: 'Loading..',
            moduleId: getResourcePage('collection')
        })
        .state('systems/:systemId/pods/:podId/measurements/:channelType/quantities', {
            route: '/systems/:systemId/pods/:podId/measurements/:channelType/quantities',
            title: 'Loading..',
            moduleId: getResourcePage('series')
        })
        .state('systems/:systemId/pods/:podId/measurements/:channelType/quantities/:quantity(/:date)', {
            route: '/systems/:systemId/pods/:podId/measurements/:channelType/quantities/:quantity(/:date)',
            title: 'Loading..',
            moduleId: getResourcePage('series')
        })
        .state('systems/:systemId/pods/:podId/profiles/:seriesType', {
            route: '/systems/:systemId/pods/:podId/profiles/:seriesType',
            title: 'Loading..',
            moduleId: getResourcePage('series')
        })
        .state('systems/:systemId/pods/:podId/:reportType/:seriesType(/:date)', {
            route: '/systems/:systemId/pods/:podId/:reportType/:seriesType(/:date)',
            title: 'Loading..',
            moduleId: getResourcePage('series')
        });

    $stateProvider.state('home').buildNavigationModel();
});

router.guardRoute = function(routeInfo, params, instance) {
    var result = authenticationService.isAuthenticated();
    if(!result) {
        result = '';
    }
    return result;
};

router.mapUnknownRoutes('contexts/client/handlers/home', '');

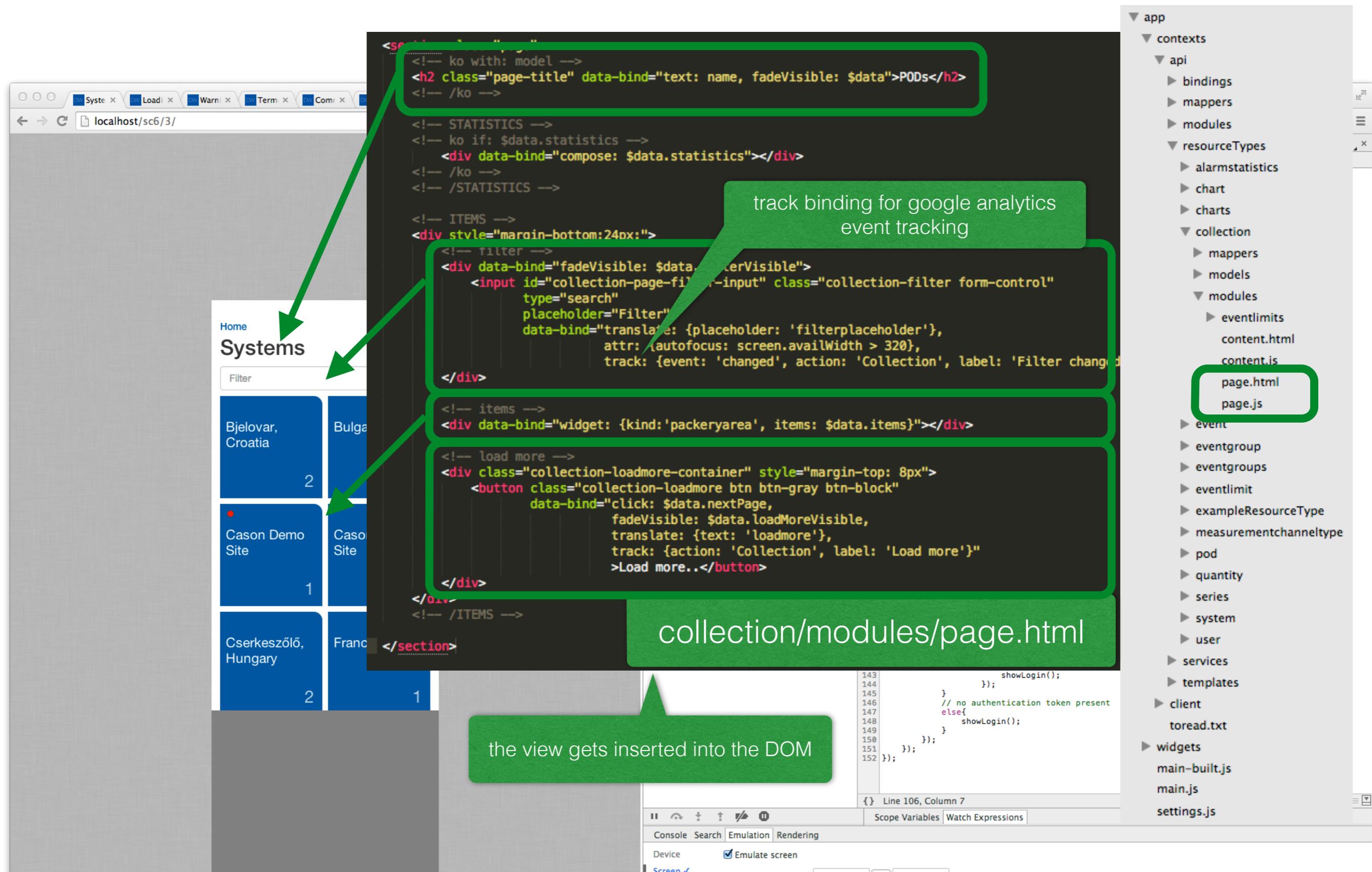
return router.activate();
});
```

Router gets activated.

app

- contexts
 - api
 - bindings
 - mappers
 - modules
 - resourceTypes
 - alarmstatistics
 - chart
 - charts
 - collection
 - mappers
 - models
 - modules
 - eventlimits
 - content.html
 - content.js
 - page.html
 - page.js
- event
- eventgroup
- eventgroups
- eventlimit
- exampleResourceType
- measurementchanneltype
- pod
- quantity
- series
- system
- user
- services
- templates
- client
- toread.txt
- widgets
- main-built.js
- main.js
- settings.js

shell.js



localhost/sc6/3/

```

define([ ... ],
  function ( ... )
{
  'use strict';

  // constructor
  var ctor = function(){ ... };

  ctor.prototype.canActivate = function(){ ... }

  ctor.prototype.activate = function (params) { ... }

  ctor.prototype.compositionComplete = function(view)
  {

  }

  ctor.prototype.detached = function(view, params)
  {
    this.pollFilterField = null;
  }

  return ctor;
}

```

called by Durandal's composition engine before activation

activation function called before the viewmodel gets bound to the view

called when the view is inserted to the dom and the viewmodel is activated and bound to it

when the view is removed from the dom

collection/modules/page.js

returns the constructor to make sure every page instance is a new instance

content.js
page.html
page.js

event
eventgroup
eventgroups
eventlimit
exampleResourceType
measurementchanneltype
pod
quantity
series
system
user
services
templates

client
toread.txt

wIDGETS
main-built.js
main.js
settings.js

Line 106, Column 7

Scope Variables Watch Expressions

Console Search Emulation Rendering

Device Emulate screen

Screen

Resolution: 1280 800

constructor. the viewmodel instance is available but **router instruction is not active yet.**

```
define([ ...  
  function (...|  
 {  
   'use strict';  
  
   //prototype scope  
  
   var ctor = function(){ ...  
  
   // instance scope  
  
   /* Statistics content */  
   this.statisticsContent = ko.observable();  
  
   /* State */  
   var filterQuery = null;  
   this.items = ko.observableArray();  
   var skip = 0;  
   var take = 20;  
   var loadMoreVisible = ko.observable(false);  
   var filterVisible = ko.observable(false);  
  
   this.model = ko.observable();  
   this.api = ko.observable();  
   this.pageResource = ko.observable();  
  
   /* Collection */  
   var currentGet = null;  
   var loadCollection = function(skip, take){  
     var query = filterQuery ? filterQuery() : null;  
     var url = currentGet ? currentGet() : null;  
     if(url){  
       var options = { skip: skip, take: take, query: query };  
       url = url + '?' + ko.toJSON(options);  
     }  
     this.model(url).bind(this);  
  
     var resetCollection = function(){  
       //console.log("reset collection");  
       this.items.removeAll();  
     }.bind(this);  
  
     var checkLoadMoreVisibility = function(model){ ...  
     }.bind(this);  
  
     var checkFilterVisibility = function(model){ ...  
     }.bind(this);  
  
     var loadItems = function(isReset, newFilterQuery){  
       console.log("loadItems", isReset, newFilterQuery);  
     }.bind(this);  
  
     var nextPage = function(){ ...  
     };  
  
     /* Filtering */  
  
     var filterResults = function(newFilterQuery){ ...  
     };  
  
     ...  
  
     this.filterResults = filterResults;  
     this.loadItems = loadItems;  
     this.loadMoreVisible = loadMoreVisible;  
     this.filterVisible = filterVisible;  
     this.nextPage = nextPage;  
   };  
};
```

model: the model representation of the page's resource ready to be bound to the view.

the api link to the resource containing the page's data.

the raw source resource of the page.
used by the parentlink widget for every page.

usual parts of page viewmodels

loads the page's resource. The collection page is a special case as it supports filtering and paging.

model vs..viewmodel:

- the model is a simple data structure without any behavior (except for data validation) mapped from the api's resource.
- the.viewmodel is a data structure tailored to meet the view's needs and contains complex behavior (like filtering and paging here)

ge.js

```
= { activate: true, tra  
el = 'contexts/client/mo  
compose(hostElement, set  
  
ce.initTranslations().al  
pp init  
actionService.isAuthenticated  
cationService.getCurrent  
tion(){  
// authentication token  
initShell();  
  
tion(){  
// authentication token  
showLogin();  
  
tication token present  
n();  
  
tions
```

ngs
ers
ales
rceTypes
rmstatistics
urt
► charts
▼ collection
► mappers
► models
▼ modules
► eventlimits
content.html
content.is
page.html
page.js
► event
► eventgroup
► eventgroups
► eventlimit
► exampleResourceType
► measurementchanneltype
► pod
► quantity
► series
► system
► user
► services
► templates
► client
toread.txt
► widgets
main-built.js
main.js
settings.js

canActivate is a great place for
redirects.

```
ctor.prototype.canActivate = function(){
    var dfd = $.Deferred();
    var result = true;

    /*
     * When a Collection page is the home page
     * then we check the Collection wether it
     * If this is the case we redirect the rou
    */
    if(apiUrlMapper.isClientHome()){
        var startupApiLink = apiUrlMapper.clientToApi('');
        dataService.get(startupApiLink, {take: -1}).then(
            function(Resource){
                if(Resource){
                    if(Resource.Type.toLowerCase() === 'collection'){
                        if(Resource.Properties.TotalCount === 1){
                            var singleItemApiLink = null;
                            for(var rel in Resource.Links){
                                if(rel.toLowerCase() !== 'parent' && rel.toLowerCase() !== 'self'){
                                    singleItemApiLink = Resource.Links[rel];
                                }
                            }
                            result = {
                                redirect: apiUrlMapper.apiToClient(singleItemApiLink)
                            };
                            dfd.resolve(result);
                            return;
                        }
                    }
                }
            }
        );
        dfd.resolve(result);
    },
    function(){
        dfd.resolve(result);
    });
}

else {
    dfd.resolve(result);
}

return dfd.promise();
}
```

redirects to the first item of the collection if the collection is the home page and it has only one item.

```
te : function(){ ... }
```

```
: function (params) { ... }
```

```
onComplete = function(view, parents){ ... }
```

```
: function(view, parent){  
= null;
```

ection/modules/page.js

```
128         var settings = { activate: true, tra
129         settings.model = 'contexts/client/mo
130         composition.compose(hostElement, set
131     );
132
133     translationService.initTranslations().al
134     //continue app init
135     if(authenticationService.isAuthenticated)
136         authenticationService.getCurrent
137         function(){
138             // authentication token
139             initShell();
140         },
141         function(){
142             // authentication token
143             showLogin();
144         };
145     // no authentication token present
146     else{
147         showLogin();
148     }
149 },);
150 },);
151 },);
152});
```

{ } Line 106, Column 7

Emulation Renderings

Emulate screen

```
▼ app
  ▼ contexts
    ▼ api
      ► bindings
      ► mappers
      ► modules
    ▼ resourceTypes
      ► alarmstatistics
      ► chart
      ► charts
      ▼ collection
        ► mappers
        ► models
      ▼ modules
        ► eventlimits
        content.html
        content.js
        page.html
        page.js
        □
      ► event
      ► eventgroup
      ► eventgroups
      ► eventlimit
      ► exampleResourceType
      ► measurementchanneltype
      ► pod
      ► quantity
      ► series
      ► system
      ► user
      ► services
      ► templates
    ► client
    toread.txt
  ► widgets
  main-built.js
  main.js
  settings.js
```

resources are standard datasets describing a realworld entity (system, pod, event, user) and linked to eachother.

generates the url to the page's resource using convention.

this url is used to fill the.viewmodel with data.

```
ctor.prototype.activate = function (params) {
    this.api(apiUrlMapper.clientToApi());

    router.on('router:navigation:complete', function(){
        pageHelpers.updateBrowserTitleToName(this.model());
    }.bind(this));

    return true;
};
```

updates browser title after navigation has completed to the page model's name.



```
define([
    'use strict';

    //prototype scope

    var ctor = function(){};

    ctor.prototype.canActivate = function(){};

    ctor.prototype.activate = function (params) {
```

```
onComplete = function(view, parents){
```

```
= function(view, parent){
    i = null;
```

Collection/modules/page.js

```
128     var settings = { activate: true, tra
129     settings.model = 'contexts/client/mo
130     composition.compose(hostElement, set
131     );
132
133     translationService.initTranslations().al
134     //continue app init
135     if(authenticationService.isAuthenticated())
136         authenticationService.getCurrent
137             function(){
138                 // authentication token
139                 initShell();
140             },
141             function(){
142                 // authentication token
143                 showLogin();
144             });
145
146     // no authentication token present
147     else{
148         showLogin();
149     };
150 };
151 });
152 });

});
```

{ } Line 106, Column 7

Scope Variables Watch Expressions

Console Search Emulation Rendering

Device Emulate screen

Screen

Resolution:

320

640

800

▼ app
 ▼ contexts
 ▼ api
 ► bindings
 ► mappers
 ► modules
 ▼ resourceTypes
 ► alarmstatistics
 ► chart
 ► charts
 ▼ collection
 ► mappers
 ► models
 ▼ modules
 ► eventlimits
 content.html
 content.js
 page.html
 page.js
 ► event
 ► eventgroup
 ► eventgroups
 ► eventlimit
 ► exampleResourceType
 ► measurementchanneltype
 ► pod
 ► quantity
 ► series
 ► system
 ► user
 ► services
 ► templates
 ► client
 toread.txt
 ► widgets
 main-built.js
 main.js
 settings.js

compositionComplete is the callback where you have **access to the view**.

```
ctor.prototype.compositionComplete = function(view, parents){  
  
    var initInlineFilter = function(){  
        var filterInputElement = document.getElementById('collection-page-filter-input');  
        // event based  
        var valueUpdateHandler = function(event) {  
            this.filterResults(filterInputElement.value);  
        }.bind(this);  
        var handler = function() { setTimeout(valueUpdateHandler, 0); }.bind(this);  
        filterInputElement.addEventListener('keydown', handler);  
  
        // plus polling based  
        this.pollFilterField = function(){  
            this.filterResults(filterInputElement.value);  
            setTimeout(this.pollFilterField, 1000);  
        }.bind(this);  
        this.pollFilterField();  
    }.bind(this);  
  
    initInlineFilter();  
};
```

```
define([  
    'use strict';  
  
    //prototype scope  
  
    var ctor = function(){  
    };  
  
    ctor.prototype.canActivate = function(){  
    };  
  
    ctor.prototype.activate = function (params) {  
    };  
  
    ctor.prototype.compositionComplete = function(view, parents){  
        var settings = { activate: true, tra  
        settings.model = 'contexts/client/mo  
        composition.compose(hostElement, set  
  
        translationService.initTranslations().al  
        //continue app init  
        if(authenticationService.isAuthenticated())  
            authenticationService.getCurrentUser()  
            function(){  
                // authentication token  
                initShell();  
            };  
        function(){  
            // authentication token  
            showLogin();  
        };  
        if(!authenticationService.isAuthenticated() &  
        login();  
    };  
};
```

here instead of value knockout binding a manual value update is used due to bugs in the value binding.

app
contexts
api
bindings
mappers
modules
resourceTypes
alarmstatistics
chart
charts
collection
mappers
models
modules
eventlimits
content.html
content.js
page.html
page.js
event
eventgroup
eventgroups
eventlimit
exampleResourceType
measurementchanneltype
pod
quantity
series
system
user
services
templates
client
toread.txt
widgets
main-built.js
main.js
settings.js

The screenshot shows a browser window displaying a 'Systems' dashboard and the source code for a module named 'page.js'. The dashboard has a grid of six items:

- Bjelovar, Croatia (2)
- Bulgaria
- Cason Demo Site (1)
- Cason's Test Site (3)
- Cserkeszőlő, Hungary (2)
- France (1)

The source code for 'page.js' is shown in the main pane:

```
define([ ... ],
  function ( ... )
{
  'use strict';

  //prototype scope

  var ctor = function(){ ... };
  ctor.prototype.canActivate = function(){ ... };

  ctor.prototype.activate = function (params) { ... };

  ctor.prototype.compositionComplete = function(view, parents){ ... };

  ctor.prototype.detached = function(view, parent){
    this.pollForChildren = null;
  };

  return ctor;
});
```

A green callout box highlights the word 'detached'.

A large green box covers the bottom right portion of the code editor, containing the text 'collection/modules/page.js'.

The right sidebar shows a tree view of the application structure:

- app
 - contexts
 - api
 - bindings
 - mappers
 - modules
 - resourceTypes
 - alarmstatistics
 - chart
 - charts
 - collection
 - mappers
 - models
 - modules
 - eventlimits
 - content.html
 - content.is
 - page.html
 - page.js
 - event
 - eventgroup
 - eventgroups
 - eventlimit
 - exampleResourceType
 - measurementchanneltype
 - pod
 - quantity
 - series
 - system
 - user
 - services
 - templates
 - client
 - toread.txt
 - widgets
 - main-built.js
 - main.js
 - settings.js

Typical Page load

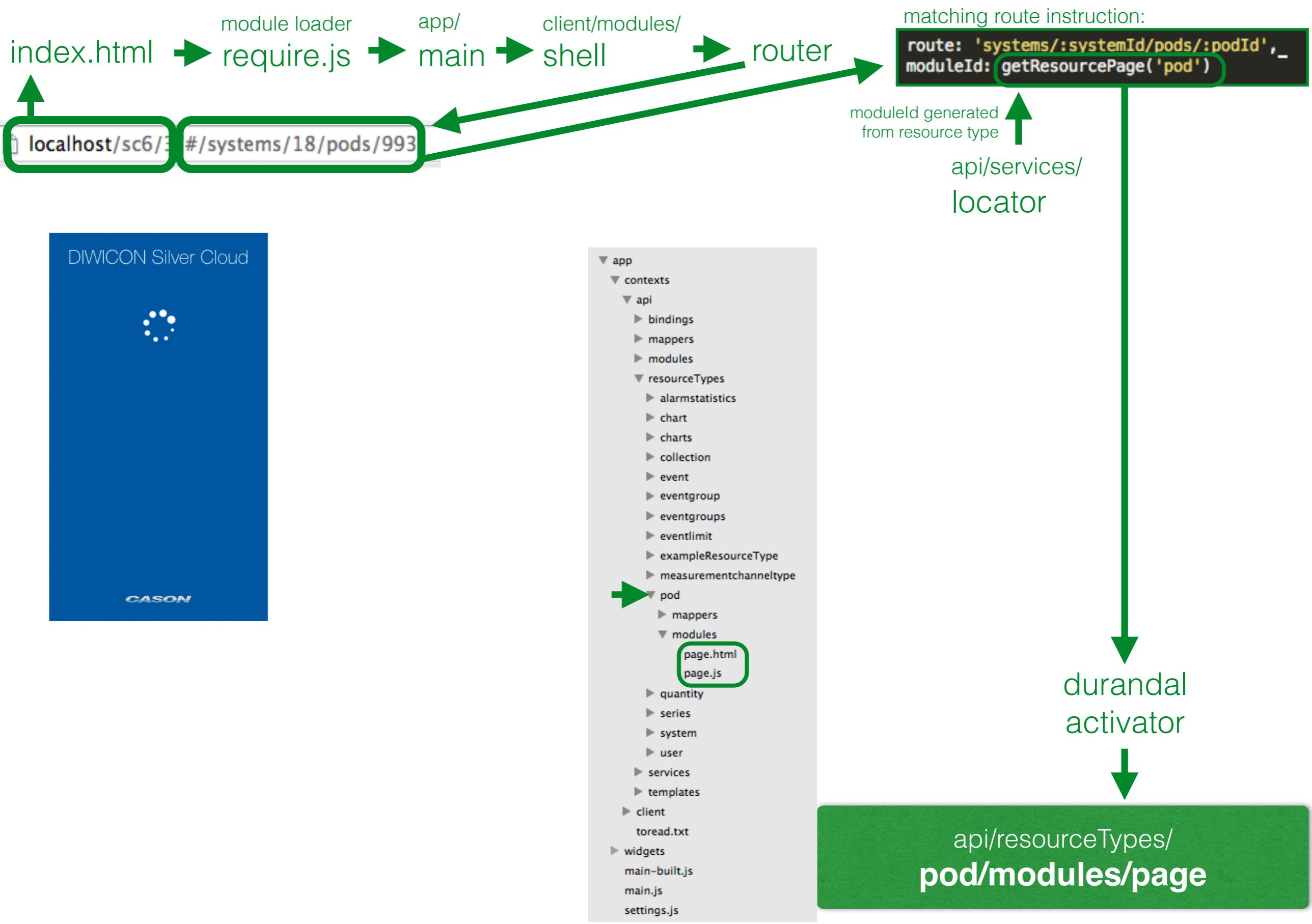
Example 2

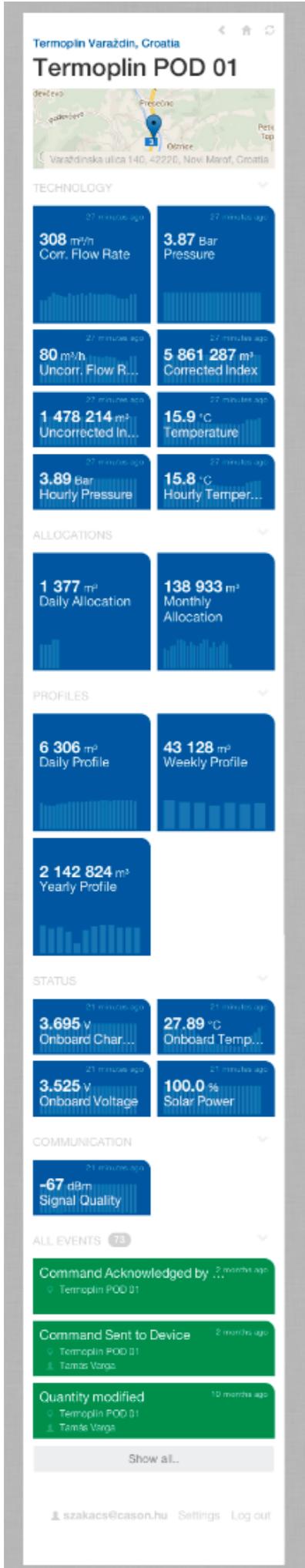
- Startup summary
- Page activation
- Client - api url conventions
- Mappers
- Dynamic content composition

Summary

- Startup
- Page activation
- Client - api url conventions
- Mappers
- Dynamic content composition







durandal activator

5. The activation will be completed when the Resource is downloaded, mapped and its model is ready.

```
1 define(['plugins/router',
2         'contexts/api/services/urlMapper',
3         'contexts/api/services/resourceMapper',
4         'contexts/client/services/dataService',
5         'contexts/client/services/pageHelper',
6         'knockout'
7     ],
8     function (
9         router,
10        apiUrlMapper,
11        resourceMapper,
12        dataService,
13        pageHelpers,
14        ko
15    ) {
16
17     return {
18         activate: function (podId) {
19
20             // saving params
21             this.podId = decodeURIComponent(podId);
22
23             this.model = ko.observable();
24
25             router.on('router:navigation:complete', function(){
26                 pageHelpers.updateBrowserTitleToName(this.model());
27             }.bind(this));
28
29             this.api = apiUrlMapper.clientToApi();
30             this.pageResource = ko.observable();
31             var gettingPromise = dataService.get(this.api, {preload: 0, needParent: true}).then(function(Resource){
32                 this.pageResource(Resource);
33                 var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34                     this.model(model);
35                     pageHelpers.updateBrowserTitleToName(model);
36                 }.bind(this));
37                 return mappingPromise;
38             }.bind(this));
39
40             // contentList
41             return gettingPromise;
42         }
43     };
44 });
45 };
```

activation will be completed when the source is loaded, and its ready.

1. An api resource gets associated with this viewmodel instance.
Convention based client path to api url mapping:
pilot.diwicon.com/sc6/#/systems/18/pods/993
pilot.diwicon.com/sc6/api/scada/systems/18/pods/993
2. Getting the associated api resource
3. Mapping the raw JSON resource to a model
4. Loading the model to an observable to be bound to the view after activation.

api/resourceTypes/
pod/modules/page.js

- I. An api resource gets associated with this viewmodel instance.

2. Getting the associated api resource

3. Mapping the raw JSON resource to a model

4. Loading the model to an observable to be bound to the view after activation.

api/resourceTypes/ **pod/modules/page.js**

The screenshot shows the Termoplín mobile application interface. At the top, it displays the location "Termoplín Varaždin, Croatia". Below this is a map showing the device's location. The main screen is divided into several sections:

- TECHNOLOGY**: Shows "308 m³/h Corr. Flow Rate" (27 minutes ago), "80 m³/h Uncorr. Flow R..." (27 minutes ago), "1 478 214 m³ Uncorrected In..." (27 minutes ago), and "3.89 Bar Hourly Pressure" (27 minutes ago).
- ALLOCATIONS**: Shows "1 377 m³ Daily Allocation" and "138 933 m³ Monthly Allocation".
- PROFILES**: Shows "6 306 m³ Daily Profile" and "43 128 m³ Weekly Profile".
- STATUS**: Shows "3.695 v Onboard Char..." and "27.89 °C Onboard Temp..." (27 minutes ago), "3.525 v Onboard Voltage" and "100.0 % Solar Power" (27 minutes ago).
- COMMUNICATION**: Shows "-67 dBm Signal Quality" (27 minutes ago).
- ALL EVENTS**: Shows recent events like "Command Acknowledged by ...", "Command Sent to Device", and "Quantity modified".

Returns a model of the raw resource in the given representation

- "card": the most important properties of the resource typically used for the cards of the resource.
- "full": a complete representation of the resource, typically used for page viewmodels of the resource..

```
    api: {
      operator
    } {
      var mapAsync = function(Resource, targetRepresentation){
        return system.defer(function(dfdf){
          if(!Resource){console.error("Resource is empty!!");return;}
          if(!Resource || !Resource.Properties || !Resource.Properties.length){console.error('Invalid Resource format');dfd.reject();return;}
          //console.log("Mapping resource", Resource);
          var model = {};
          var basicMapping = function(){...};
          var nameMapping = function(model){...};
          var typeSpecificMapping = function(){...};
          basicMapping()
            .then(nameMapping)
            .then(typeSpecificMapping);
          return new Promise(function(resolve, reject){model.promise().then(resolve).catch(reject);});
        });
        return {
          toModel : mapAsync,
          map: mapAsync
        };
      };
    }
  };
}
```

1. basic mapping: mapping the model properties that are standard for all models (like name)

2. type specific mapping: loading the resource type specific mapper with the module loader. the type specific mapper will return the resource type specific model.

contexts/api/services/
resourceMapper

```
33   var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38   }.bind(this));
39
40   // contentList
41
42   return gettingPromise;
43 }
```

api/resourceTypes/
pod/modules/page.js

The screenshot shows the Termoplín PC software interface. At the top left, it displays 'Termoplín Varaždin, Croatia' with a map of the location. Below the map are sections for 'TECHNOLOGY' (Corr. Flow Rate, Uncorr. Flow Rate, Unorrected In., Hourly Pressure), 'ALLOCATIONS' (Daily Allocation, Daily Profile, Yearly Profile), 'STATUS' (Onboard Char., Onboard Voltage, Solar Power), 'COMMUNICATION' (Signal Quality), and 'ALL EVENTS' (Command Acknowledged by, Command Sent to Device, Quantity modified). The 'app' folder structure on the right includes 'contexts', 'api' (bindings, mappers, modules, resourceTypes), and 'services' (authentication.js, locator.js, resourceMapper.js).

```

define([
  ...
], [
  ...
], function (...) {
  'use strict';

  var contentContainerCompare = function(a,b){ ... };

  var linkedContentMetadata = { ... };

  //TODO it wont show content if is embedded
  // TODO do not mix presentation (e.g. content containers) with models (e.
  // TODO support no metadata with defaults
  //TODO review the role of containers and make a generic getLinkedViewMode
  var getLinkedContentContainer = function(Resource, rel, metadata){ ... };

  var getLinkedContentContainers = function(Resource, metadata){ ... };

  var toModel = function(Resource, model, targetRepresentation){ ... };

  return {
    toModel: toModel
  };
});

```

api/resourceTypes/
pod/mappers/mapper

```

var typeSpecificMapping = function(){ ... };

basicMapping()
  .then(nameMapping)
  .then(typeSpecificMapping);

}).promise();

};

return {
  toModel : mapAsync,
  map: mapAsync
};
});

```

2. type specific mapping: loading the resource type specific mapper with the module loader. the type specific mapper will return the resource type specific model.

contexts/api/services/
resourceMapper

```

33   var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38   }.bind(this));

39   // contentList

40   return gettingPromise;
41
42
43 }

```

api/resourceTypes/
pod/modules/page.js

The dashboard displays various metrics and configurations for the Termoplín POD 01. Key sections include:

- TECHNOLOGY**: Shows Corr. Flow Rate (308 m³/h), Pressure (3.87 Bar), Uncorr. Flow R... (80 m³/h), Corrected Index (5 861 287 m³), Uncorrected In... (1 478 214 m³), Temperature (15.9 °C), Hourly Pressure (3.89 Bar), and Hourly Temper... (15.8 °C).
- ALLOCATIONS**: Daily Allocation (1 377 m³) and Monthly Allocation (138 933 m³).
- PROFILES**: Daily Profile (6 306 m³), Weekly Profile (43 128 m³), and Yearly Profile (2 142 824 m³).
- STATUS**: Onboard Char... (3.695 V), Onboard Temp... (27.89 °C), Onboard Voltage (3.525 V), and Solar Power (100.0 %).
- COMMUNICATION**: Signal Quality (-67 dBm).
- ALL EVENTS**: A list of recent events including Command Acknowledged by ... (2 months ago), Command Sent to Device (2 months ago), and Quantity modified (10 months ago).

```

define([
  ...
], [
  ...
], [
  ...
])
  'use strict';

  var contentContainerCompare = function(a,b){ ... };

  var linkedContentMetadata = { ... };

  //TODO it wont show content if is embedded
  // TODO do not mix presentation (e.g. content containers) with models (e.
  // TODO support no metadat with defaults
  //TODO review the role of containers and make a generic getLinkedViewMode
  var getLinkedContentContainer = function(Resource, rel, metadata){ ... };

  var getLinkedContentContainers = function(Resource, metadata){ ... };

  var toModel = function(Resource, model, targetRepresentation){ ... };

  return {
    toModel: toModel
};

});
```

api/resourceTypes/
pod/mappers/mapper

```

33   var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38   }.bind(this));

39   // contentList

40   return gettingPromise;
41 }

42 }

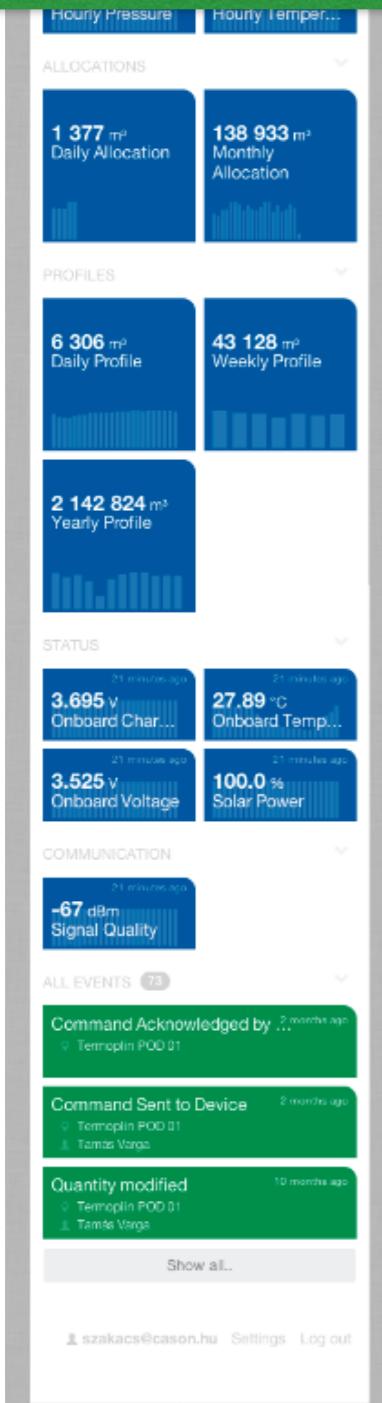
43 }
```

api/resourceTypes/
pod/modules/page.js

```
// representation specific mapping
if(targetRepresentation === "full"){

  model.contentList = ko.observableArray();
  getLinkedContentContainers(Resource, linkedContentMetadata).then(
    function(linkedContentContainers){
      linkedContentContainers.sort(contentContainerCompare);
      linkedContentContainers.map(function(result){
        model.contentList.push(result);
      });
      dfd.resolve(model);
    },
    function(){
      console.error("ERROR");
    }
  );
}
```

api/resourceTypes/pod/mappers/mapper



The list of modules serving as the content of the page (All Events collection content, Profiles collection content, Technology Measurements collection content, etc.)

```
targetMetadata = { ... }

// show content if is embedded
// mix presentation (e.g. content containers) with models (e.g.
// no metadata with defaults
// the role of containers and make a generic getLinkedViewMode
ContentContainer = function(Resource, rel, metadata){ ... }

var getLinkedContentContainers = function(Resource, metadata){ ... };

var toModel = function(Resource, model, targetRepresentation){ ... };

return {
  toModel: toModel
};
```

```
33   var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38   }.bind(this));

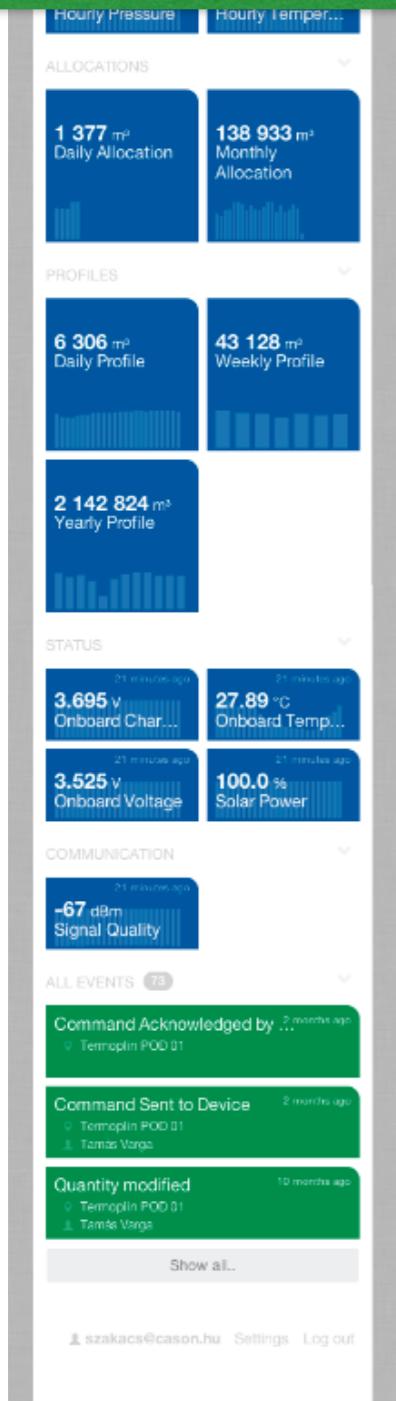
39   // contentList
40
41   return gettingPromise;
42
43 }
```

api/resourceTypes/
pod/modules/page.js

```
// representation specific mapping
if(targetRepresentation === "full"){
    model.contentList = getLinkedContentContainers();
    getLinkedContentContainers(Resource, linkedContentMetadata).then(function(linkedContentContainers){
        linkedContentContainers.sort(contentContainerCompare);
        linkedContentContainers.map(function(result){
            model.contentList.push(result);
        });
        dfd.resolve(model);
    }),function(){
        console.error("ERROR");
    }
}

```

api/resourceTypes/pod/mappers/mapper



The list of modules serving as the content of the page (All Events collection content, Profiles collection content, Technology Measurements collection content, etc.)

The **content list is obtained automatically** from the pod resource using a **metadata descriptor** of which of its links should serve as content and in which order.

```
var getLinks = function(Resource, targetRepresentation) {
    var toModel = function(Resource, model, targetRepresentation) {
        return {
            toModel: toModel
        };
    };
    return {
        getLinks: getLinks,
        toModel: toModel
    };
};
```

```
var linkedContentMetadata = {
    Measurements: {
        substitueWithLinks: true,
        linkedMetadata: {
            Technology: { ... },
            Status: { ... },
            Communication: { ... }
        }
    },
    Allocations: {
        order: 1,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Nominations: {
        order: 2,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Profiles: {
        order: 3,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Events: {
        getParams: { preload: 1 },
        substitueWithLinks: true,
        linkedMetadata: {
            Alarms: { ... },
            Warnings: { ... },
            All: { ... }
        }
    }
};

/*,
AlarmStatistics: {
    order: 5,
    isImportant: true
}*/
```

api/resourceTypes/pod/mappers/mapper

```
33 var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38     }.bind(this));
39
40 // contentList
41
42     return gettingPromise;
43 }
```

api/resourceTypes/pod/modules/page.js

```
// representation specific mapping
if(targetRepresentation === "full"){
    model.contentList = getLinkedContentContainers();
    getLinkedContentContainers(Resource, linkedContentMetadata).then(function(linkedContentContainers){
        linkedContentContainers.sort(contentContainerCompare);
        linkedContentContainers.map(function(result){
            model.contentList.push(result);
        });
        dfd.resolve(model);
    }, function(){
        console.error("ERROR");
    });
}

```

api/resourceTypes/pod/mappers/mapper



```
var representation = 'content';
var contentPromise = resourceToViewModelService.getViewModelPromise(
    Resource.Links[rel],
    Resource.Embedded[rel],
    representation,
    metadata.getParams());

```

```
var title = translationService.translateResourceProperty(Resource, Resource.Tr
var linkedContentContainer = new ContentContainer(
    title,
    !metadata.isImportant,
    contentPromise,
    metadata.order);

dfd.resolve([linkedContentContainer]);
```

api/resourceTypes/pod/mappers/mapper

The list of modules serving as the content of the page (All Events collection content, Profiles collection content, Technology Measurements collection content, etc.)

The **content list is obtained automatically** from the pod resource using a **metadata descriptor** of which of its links should serve as content and in which order.

```
var getLi
};

var toModel = function(Resource, model, targetRepresentatio
};


```

For every resource marked by the metadata as content its **module with 'content' content type is dynamically obtained** from the resourceToViewModel service.

```
var linkedContentMetadata = {
    Measurements: {
        substitueWithLinks: true,
        linkedMetadata: {
            Technology: { ... },
            Status: { ... },
            Communication: { ... }
        }
    },
    Allocations: {
        order: 1,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Nominations: {
        order: 2,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Profiles: {
        order: 3,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Events: {
        getParams: { preload: 1 },
        substitueWithLinks: true,
        linkedMetadata: {
            Alarms: { ... },
            Warnings: { ... },
            All: { ... }
        }
    }
};
```

```
/*
    AlarmStatistics: {
        order: 5,
        isImportant: true
    }
*/
```

api/resourceTypes/pod/mappers/mapper

```
33
34
35
36
37
38
39
40
41
42
43
```

```
var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
    this.model(model);
    pageHelpers.updateBrowserTitleToName(model);
}).bind(this));
return mappingPromise;
}.bind(this));

// contentList

return gettingPromise;
```

api/resourceTypes/pod/modules/page.js

```
// representation specific mapping
if(targetRepresentation === "full"){
    model.contentList = getLinkedContentContainers();
    getLinkedContentContainers(Resource, linkedContentMetadata).then(function(linkedContentContainers){
        linkedContentContainers.sort(contentContainerCompare);
        linkedContentContainers.map(function(result){
            model.contentList.push(result);
        });
        dfd.resolve(model);
    }, function(){
        console.error("ERROR");
    });
}

```

api/resourceTypes/pod/mappers/mapper



```
var representation = 'content';
var contentPromise = resourceToViewModelService.getViewModelPromise(
    Resource.Links[rel],
    Resource.Embedded[rel],
    representation,
    metadata.getParams());

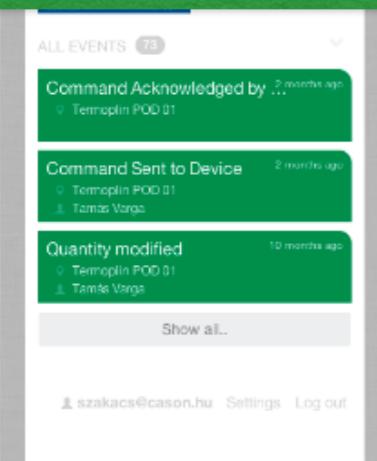
var linkedContentContainer = new ContentContainer(
    title,
    !metadata.isImportant,
    contentPromise,
    metadata.order);

dfd.resolve([linkedContentContainer]);

```

api/resourceTypes/pod/mappers/mapper

This dynamic content is added to the generic content model, and pushed to the contentList of the pod model.



33
34
35
36
37
38
39
40
41
42
43

```
var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
    this.model(model);
    pageHelpers.updateBrowserTitleToName(model);
}).bind(this));
return mappingPromise;
}.bind(this));

// contentList

return gettingPromise;
}
```

api/resourceTypes/
pod/modules/page.js

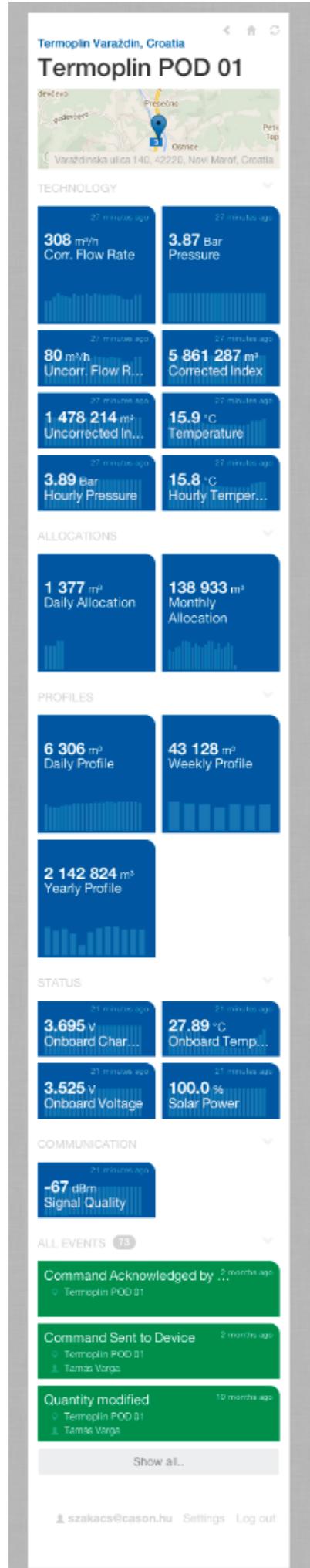
The list of modules serving as the content of the page (All Events collection content, Profiles collection content, Technology Measurements collection content, etc.)

The **content list is obtained automatically** from the pod resource using a **metadata descriptor** of which of its links should serve as content and in which order.

```
var linkedContentMetadata = {
    Measurements: {
        substitueWithLinks: true,
        linkedMetadata: {
            Technology: { ... },
            Status: { ... },
            Communication: { ... }
        }
    },
    Allocations: {
        order: 1,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Nominations: {
        order: 2,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Profiles: {
        order: 3,
        isImportant: true,
        getParams: { preload: 1 }
    },
    Events: {
        getParams: { preload: 1 },
        substitueWithLinks: true,
        linkedMetadata: {
            Alarms: { ... },
            Warnings: { ... },
            All: { ... }
        }
    }
};

/*,
AlarmStatistics: {
    order: 5,
    isImportant: true
}*/
```

api/resourceTypes/pod/mappers/mapper



```

define([
  ...
], [
  function (...) {
    'use strict';

    var contentContainerCompare = function(a,b){ ... };
    var linkedContentMetadata = { ... };

    //TODO it wont show content if is embedded
    // TODO do not mix presentation (e.g. content containers) with models (e.
    // TODO support no metadat with defaults
    //TODO review the role of containers and make a generic getLinkedViewMode
    var getLinkedContentContainer = function(Resource, rel, metadata){ ... };

    var getLinkedContentContainers = function(Resource, metadata){ ... };

    var toModel = function(Resource, model, targetRepresentation){ ... };

    return {
      toModel: toModel
    };
  }
]);
  
```

api/resourceTypes/
pod/mappers/mapper

After the content list with the dynamic content modules are ready the model is resolved.

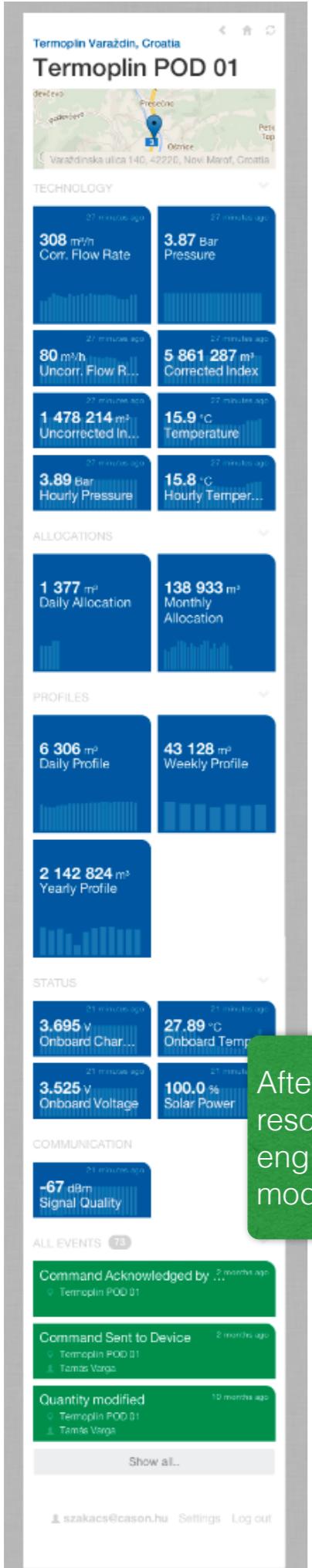
pod full model

```

33   var mappingPromise = resourceMapper.toModel(Resource).then(function(model){
34     this.model(model);
35     pageHelpers.updateBrowserTitleToName(model);
36     }.bind(this));
37     return mappingPromise;
38   }.bind(this));

39   // contentList
40
41   return gettingPromise;
42
43 }
  
```

api/resourceTypes/
pod/modules/page.js



```
<section class="page">
  <!-- ko with: model -->
  <h2 class="page-title" data-bind="fadeVisible: $data">
    <!-- ko if: isInAlarm || isInWarning -->
    <span data-bind="style: {color: $data.statusColor}"></span>
    <!-- /ko -->
    <span data-bind="text: name"></span>
  </h2>

  <!-- ko with: location -->
  <!-- ko widget -->
    kind: 'map',
    widthHeightRatio: 3,
    longitude: longitude, latitude: latitude,
    markerColor: $parent.statusColor
  <!-- /ko -->

  <!-- /ko -->

  <div data-bind="widget: location">
    kind: 'expander',
    items:contentList,
    headerProperty:'title',
    isClosedProperty:'isClosed'
  <!-- /ko -->
</section>
```

api/resourceTypes/
pod/modules/page.html

After the activation promise is resolved the Durandal composition engine loads and binds the module's view.

pod full model

```
34   mappingPromise = resourceMapper.toModel(Resource).then(function(model){
35     this.model(model);
36     pageHelpers.updateBrowserTitleToName(model);
37     }.bind(this));
38     return mappingPromise;
39     }.bind(this));
40
41     // contentList
42
43     return gettingPromise;
44 }
```

api/resourceTypes/
pod/modules/page.js

With Knockout's with directive we scope the binding context from the whole.viewmodel to the.viewmodel's model property.

The screenshot shows a complex web-based dashboard for the Termoplín POD 01. On the left, there's a map of Varaždin, Croatia, with a location marker. Below the map are several sections: TECHNOLOGY, ALLOCATIONS, PROFILES, STATUS, COMMUNICATION, and ALL EVENTS. The TECHNOLOGY section contains cards for Corr. Flow Rate, Pressure, Uncorr. Flow R..., Corrected Index, Uncorrected In..., Temperature, Hourly Pressure, and Hourly Temper... The ALLOCATIONS section shows Daily Allocation and Monthly Allocation. The PROFILES section displays Daily Profile, Weekly Profile, and Yearly Profile. The STATUS section includes Onboard Char... and Onboard Temp... The COMMUNICATION section shows Signal Quality. The ALL EVENTS section lists Command Acknowledged by..., Command Sent to Device, and Quantity modified.

```

<section class="page">
    <!-- ko with: model -->
    <h2 class="page-title" data-bind="fadeVisible: $data">
        <!-- ko if: isInAlarm || isInWarning -->
        <span data-bind="style: {color: $data.statusColor}"></span>
        <!-- /ko -->
        <span data-bind="text: name"></span>
    </h2>

    <!-- ko with: location -->
    <!-- ko widget: {
        kind: 'map',
        widthHeightRatio: 3,
        longitude: longitude, latitude: latitude,
        markerColor: $parent.statusColor
    } --><!-- /ko -->

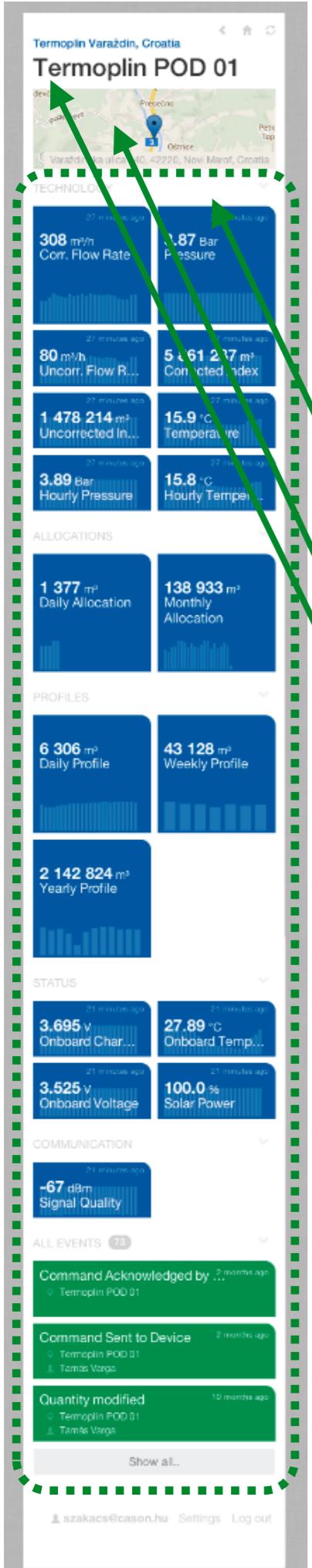
    <!-- /ko -->

    <div data-bind="widget: {
        kind: 'expander',
        items:contentList,
        headerProperty:'title',
        isClosedProperty:'isClosed'
    }"></div>
    <!-- /ko -->
</section>

```

With Knockout's with directive we scope the binding context from the whole viewmodel to the viewmodel's model property.

api/resourceTypes/
pod/modules/page.html



```

<section class="page">
  <!-- ko with: model -->
  <h2 class="page-title" data-bind="fadeVisible: $data">
    <!-- ko if: isInAlarm || isInWarning -->
    <span data-bind="style: {color: $data.statusColor}"></span>
    <!-- /ko -->
    <span data-bind="text: name"></span>
  </h2>

  <!-- ko with: location -->
  <!-- ko widget: {
    kind: 'map',
    widthHeightRatio: 3,
    longitude: longitude, latitude: latitude,
    markerColor: $parent.statusColor
  } --><!-- /ko -->

  <!-- /ko -->

  <div data-bind="widget: {
    kind: 'expander',
    items:contentList,
    headerProperty:'title',
    isClosedProperty:'isClosed'
  }"></div>
</section>

```

api/resourceTypes/
pod/modules/pod01.html

```

  "apiSelf": "https://pilot.diwicon.com/sc6/api/scada/systems/18",
  "cardView": "contexts/api/templates/card.pod",
  "clientUrl": "#/systems/18/pods/993",
  "contentList": Array[9],
  "displayOrder": null,
  "isImportant": true,
  "isInAlarm": false,
  "isInWarning": false,
  "location": Object,
    address: "Varaždinska ulica 140, 42220, Novi Marof, Croatia",
    latitude: 46.18897,
    longitude: 16.3452,
    name: "Termoplín POD 01",
    > __proto__: Object,
    "name": "Termoplín POD 01",
    "representation": "full",
    > "resource": Object,
    "resourceType": "Pod",
    "status": "Normal",
    "statusColor": "#00529E"
  }
}

```

pod full model

With Knockout's with directive we scope the binding context from the whole viewmodel to the viewmodel's model property.

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Pow...

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
2 months ago
Termoplín POD 01

Command Sent to Device
2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified
10 months ago
Termoplín POD 01
Tamás Varga

Show all..

```
<div data-bind="widget: {  
    kind: 'expander'  
    items: contentL  
    headerProperty:  
    isClosedProper  
}"></div>
```

```
<div class="accordion" data-bind="foreach: { data: settings.items,  
    <!-- ko if: $data!==undefined && $data!==null && !$data.isHidden() -->  
<div class="accordion-group">  
    <div class="accordion-heading non-selectable">...</div>  
    <div class="accordion-body collapse in">  
        <div class="accordion-inner" data-part="itemContainer" >  
            <!-- ko if: $data.errorMessage -->  
            <div data-part="error-message">  
                <p data-bind="translate: {text: $data.errorMessage}, visible: $data.errorMessage</p>  
            </div>  
            <!-- /ko -->  
            <div data-part="item">  
                <!-- if: !isClosed() -->  
                <div data-bind="widget: {  
                    kind: 'animatedcompositionhost',  
                    content: content,  
                    id: content() ? content().apiLink : null,  
                    isLoading: isLoading}></div>  
                <!-- / -->  
            </div>  
        </div>  
    </div>  
</div>
```

widgets/expander/view.html

api/resourceTypes/
pod/modules/pod.html

```
" apiSelf": "https://pilot.diwicon.com/sc6/api/scada/systems/18/pods/993",  
" cardView": "contexts/api/templates/card.pod",  
" clientUrl": "#/systems/18/pods/993",  
" contentList": Array[9],  
" displayOrder": null,  
" isImportant": true,  
" isInAlarm": false,  
" isInWarning": false,  
" location": Object  
    address: "Varaždinska ulica 140, 42220, Novi Marof, Croatia"  
    latitude: 46.18897  
    longitude: 16.3452  
    name: "Termoplín POD 01"  
    > __proto__: Object  
    " name": "Termoplín POD 01"  
    " representation": "full",  
    > " resource": Object  
    " resourceType": "Pod",  
    " status": "Normal",  
    " statusColor": "#00529E",  
    " type": "Pod"
```

pod full model

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
2 months ago
Termoplín POD 01

Command Sent to Device
2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified
10 months ago
Termoplín POD 01
Tamás Varga

Show all..

With Konckout's foreach directive for every item in the array given in the expander widget's items settings will be rendered with the template in the element.

```
<div data-bind="widget: {  
    kind: 'expander',  
    items:contentList,  
    headerProperty: 'title',  
    isClosedProperty: 'isClose'  
}"></div>
```

```
<div class="accordion" data-bind="foreach: { data: settings.items,  
    as: 'item' }>  
    <div class="accordion-group">  
        <div class="accordion-heading non-selectable">...</div>  
        <div class="accordion-body collapse in">  
            <div class="accordion-inner" data-part="itemContainer" >  
                <!-- ko if: $data.errorMessage -->  
                <div data-part="error-message">  
                    <p data-bind="translate: {text: $data.errorMessage}, visible: $data.errorMessage != null" />  
                </div>  
                <!-- /ko -->  
                <div data-part="item">  
                    <!-- if: !isClosed() -->  
                    <div data-bind="widget: {  
                        kind: 'animatedcompositionhost',  
                        content: content,  
                        id: content() ? content().apiLink : null,  
                        isLoading: isLoading}"></div>  
                <!-- / -->  
            </div>  
        </div>  
    </div>  
</div>
```

widgets/expander/view.html

api/resourceTypes/pod/modules/page.html

```
" apiSelf": "https://api.oceanus.com/api/resourceTypes/pod/modules/page.html?podId=18&pageId=993",  
" cardView": false,  
" clientUrl": "#/systems/18/pods/993",  
" contentList": Array[9],  
" displayOrder": null,  
" isImportant": true,  
" isInAlarm": false,  
" isInWarning": false,  
" location": Object  
    address: "Varaždinska ulica 140, 42220, Novi Marof, Croatia"  
    latitude: 46.18897  
    longitude: 16.3452  
    name: "Termoplín POD 01"  
    > __proto__: Object  
" name": "Termoplín POD 01",  
" representation": "full",  
" resource": Object  
" resourceType": "Pod",  
" status": "Normal",  
" statusColor": "#00529E"
```

pod full model

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Power

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
2 months ago
Termoplín POD 01

Command Sent to Device
2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified
10 months ago
Termoplín POD 01
Tamás Varga

Show all..

With Knockout's foreach directive for every item in the array given in the expander widget's items settings will be rendered with the template in the element.

```
<div class="accordion" data-bind="foreach: { data: settings.items, as: 'item' }>
  <div class="accordion-group">
    <div class="accordion-heading non-selectable"> ...
    </div>

    <div class="accordion-body collapse in">
      <div class="accordion-inner" data-part="itemContainer" >
        <!-- ko if: $data.errorMessage -->
        <div data-part="error-message">
          <p data-bind="translate: {text: $data.errorMessage}, if: !$data.errorMessage" />
        </div>
        <!-- /ko -->
        <div data-part="item">
          <!-- if: !isClosed() -->
          <div data-bind="widget: { kind: 'animatedcompositionhost', content: content, id: content() ? content().apiLink : null, isLoading: isLoading}">
            <!-- / -->
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

widgets/expander/view.html

```
" apiSelf": "https://pilot.diwicon.com/sc6/api/scada/systems/18",
" cardView": "contexts/api/templates/card.pod",
" clientUrl": "#/systems/18/pods/993"
  ▶ " contentList": Array[9]
    & displayOrder: null
    " isImportant": true
    " isInAlarm": false
    " isInWarning": false
  ▶ " location": Object
    address: "Varaždinska ulica 140, 42220, Novi Marof, Croatia"
    latitude: 46.18897
    longitude: 16.3452
    name: "Termoplín POD 01"
  ▶ __proto__: Object
    " name": "Termoplín POD 01"
    " representation": "full"
  ▶ " resource": Object
    " resourceType": "Pod"
    " status": "Normal"
    " statusColor": "#00529E"
    " type": "Pod"
```

pod full model

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Power

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
2 months ago
Termoplín POD 01

Command Sent to Device
2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified
10 months ago
Termoplín POD 01
Tamás Varga

Show all..

With Knockout's foreach directive for every item in the array given in the expander widget's items settings will be rendered with the template in the element.

```
<div class="accordion" data-bind="foreach: { data: settings.items, as: 'item' }>
  <div class="accordion-group">
    <div class="accordion-heading non-selectable"> ...
    </div>

    <div class="accordion-body collapse in">
      <div class="accordion-inner" data-part="itemContainer" >
        <!-- ko if: $data.errorMessage -->
        <div data-part="error-message">
          <p data-bind="translate: {text: $data.errorMessage}, if: !$data.errorMessage" />
        </div>
        <!-- /ko -->
        <div data-part="item">
          <!-- if: !isClosed() -->
          <div data-bind="widget: { kind: 'animatedcompositionhost', content: content, id: content() ? content().apiLink : null, isLoading: isLoading}">
            <!-- / -->
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

widgets/expander/view.html

```
" apiSelf": "https://pilot.diwicon.com/sc6/api/scada/systems/18",
" cardView": "contexts/api/templates/card.pod",
" clientUrl": "#/systems/18/pods/993"
  ▶ " contentList": Array[9]
    & displayOrder: null
    " isImportant": true
    " isInAlarm": false
    " isInWarning": false
  ▶ " location": Object
    address: "Varaždinska ulica 140, 42220, Novi Marof, Croatia"
    latitude: 46.18897
    longitude: 16.3452
    name: "Termoplín POD 01"
  ▶ __proto__: Object
    " name": "Termoplín POD 01"
    " representation": "full"
  ▶ " resource": Object
    " resourceType": "Pod"
    " status": "Normal"
    " statusColor": "#00529E"
    " type": "Pod"
```

pod full model

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Power

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
2 months ago
Termoplín POD 01

Command Sent to Device
2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified
10 months ago
Termoplín POD 01
Tamás Varga

Show all..

With Knockout's foreach directive for every item in the array given in the expander widget's items settings will be rendered with the template in the element.

```
<div class="accordion" data-bind="foreach: { data: settings.items, as: 'item' }>
```

```
    <div class="accordion-group">
```

```
        <div class="accordion-heading non-selectable">...
```

```
        </div>
```

```
        <div class="accordion-body collapse in">
```

```
            <div class="accordion-inner" data-part="itemContainer" >
```

```
                <!-- ko if: $data.errorMessage -->
```

```
                <div data-part="error-message">
```

```
                    <p data-bind="translate: {text: $data.errorMessage}, style: {color: $data.errorColor}">...
```

```
                </div>
```

```
                <!-- /ko -->
```

```
                <div data-part="item">
```

```
                    <!-- if: !isClosed() -->
```

```
                    <div data-bind="widget: {
```

```
                        kind: 'animatedcompositionhost',
```

```
                        content: content,
```

```
                        id: content() ? content().apiLink : null,
```

```
                        isLoading: isLoading}"></div>
```

```
        <!-- contentList: Array[9] -->
```

```
        <!-- 0: Object -->
```

```
        <!-- 1: Object -->
```

```
        <!-- 2: Object -->
```

```
        <!-- 3: Object -->
```

```
        <!-- 4: Object -->
```

```
        <!-- moduleId: "contexts/client/models/content" -->
```

```
        <!-- content: Object -->
```

```
        <!-- count: null -->
```

```
        <!-- errorMessage: undefined -->
```

```
        <!-- isClosed: false -->
```

```
        <!-- isForbidden: false -->
```

```
        <!-- isHidden: false -->
```

```
        <!-- isInError: false -->
```

```
        <!-- isLoading: false -->
```

```
        <!-- order: 4 -->
```

```
        <!-- title: "Status" -->
```

```
        <!-- content: Object -->
```

```
        <!-- 5: Object -->
```

```
        <!-- 6: Object -->
```

```
        <!-- 7: Object -->
```

```
        <!-- 8: Object -->
```

```
        <!-- moduleId: "contexts/client/models/content" -->
```

```
        <!-- content: Object -->
```

```
        <!-- count: 73 -->
```

```
        <!-- errorMessage: undefined -->
```

```
        <!-- isClosed: false -->
```

```
        <!-- isForbidden: false -->
```

```
        <!-- isHidden: false -->
```

```
        <!-- isInError: false -->
```

```
        <!-- isLoading: false -->
```

```
        <!-- order: 8 -->
```

```
        <!-- title: "All Events" -->
```

```
        <!-- __proto__: Object -->
```

```
        <!-- length: 9 -->
```

Widgets/expander/view.html

api/scada/systems/18/f"

pod full model.contentList

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Power

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ... 2 months ago
Termoplín POD 01

Command Sent to Device 2 months ago
Termoplín POD 01
Tamás Varga

Quantity modified 10 months ago
Termoplín POD 01
Tamás Varga

Show all..

With Konckout's foreach directive for every item in the array given in the expander widget's items settings will be rendered with the template in the element.

```
<div class="accordion" data-bind="foreach: { data: settings.items, as: 'item' }>
  <div class="accordion-group">
    <div class="accordion-heading non-selectable">...
```

▼ 4: Object
 __moduleId__: "contexts/client/models/content"
 content: Object
 count: null
 errorMessage: undefined
 isClosed: false
 isForbidden: false
 isHidden: false
 isInError: false
 isLoading: false
 order: 4
 title: "Status"

widgets/expander/view.html

pod full model.contentList

2 142 824 m³
Yearly Profile



STATUS

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
Termoplín POD 01

Command Sent to Device
Termoplín POD 01
Tamás Varga

Quantity modified
Termoplín POD 01
Tamás Varga

Show all..

data

template

```
<div class="accordion" data-bind="foreach: { data: settings.items, as: 'item' }>
  <div class="accordion-group">
    <div class="accordion-heading non-selectable">...
```

contexts/expander/view.html

```
▼ 4: Object
  moduleId : "contexts/client/models/content"
  ▼ content: Object
    moduleId : "contexts/api/resourceTypes/content"
    activate: function () {
      apiLink: "https://pilot.diwicon.com/sc6/api/resourceTypes/content"
      areDetailsClosed: undefined
      collectionLink: undefined
    }
    map: function () { [native code] }
    model: Object
      name: undefined
    sourceResource: Object
    ...proto...
    count: null
    errorMessage: undefined
    isClosed: false
    isForbidden: false
    isHidden: false
    isInError: false
    isLoading: false
    order: 4
    title: "Status"
```

pod full **model.contentList**

2 142 824 m³
Yearly Profile



STATUS



data

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
Termoplín POD 01

Command Sent to Device
Termoplín POD 01
Tamás Varga

Quantity modified
Termoplín POD 01
Tamás Varga

Show all..

template

```
<!-- if: isClosed() -->
<div data-bind="widget: {
    kind: 'animatedcompositionhost',
    content: content,
    id: content() ? content().apiLink : null,
    isLoading: isLoading}"></div>
```

widgets/
animatedcompositionhost

```
content: Object
__moduleId__: "contexts/api/resourceTypes/co
▶ activate: function () {
  apiLink: "https://pilot.diwicon.com/sc6/api/
  areDetailsClosed: undefined
  collectionLink: undefined
  ▶ map: function () { [native code] }
  ▶ model: Object
  name: undefined
  ▶ sourceResource: Object
```

pod full **model.contentList[4]**

inserts content setting.
remembers last size (local storage).
animates to size changes.

2 142 824 m³
Yearly Profile

STATUS

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
Termoplín POD 01

Command Sent to Device
Termoplín POD 01
Tamás Varga

Quantity modified
Termoplín POD 01
Tamás Varga

Show all..

Durandal's compose binding: inserts the module defined by its.viewmodel to the DOM.

widgets/
animatedcompositionhost/view.html

```
<div>
  <div data-bind="compose: {model: settings.content}"></div>
<div>
```

template

```
<!-- if: isClosed() -->
<div data-bind="widget: {
  kind: 'animatedcompositionhost',
  content: content,
  id: content() ? content().apiLink : null,
  isLoading: isLoading}"></div>
```

widgets/
animatedcompositionhost

data

```
▼ content: Object
  __moduleId__: "contexts/api/resourceTypes/co
▶ activate: function () {
  apiLink: "https://pilot.diwicon.com/sc6/api/
  areDetailsClosed: undefined
  collectionLink: undefined
▶ map: function () { [native code] }
▶ model: Object
  name: undefined
▶ sourceResource: Object
```

pod full model.contentList[4]

2 142 824 m³
Yearly Profile

STATUS

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
Termoplín POD 01

Command Sent to Device
Termoplín POD 01
Tamás Varga

Quantity modified
Termoplín POD 01
Tamás Varga

Show all..

Durandal's compose binding: inserts the module defined by its.viewmodel to the DOM.

widgets/
animatedcompositionhost/view.html

```
<div>
  <div data-bind="compose: {model: settings.content}"></div>
<div>
```

template

```
<!-- if: isClosed() -->
<div data-bind="widget: {
  kind: 'animatedcompositionhost',
  content: content,
  id: content() ? content().apiLink : null,
  isLoading: isLoading}"></div>
```

widgets/
animatedcompositionhost

data

```
▼ content: Object
  __moduleId__: "contexts/api/resourceTypes/co
▶ activate: function () {
  apiLink: "https://pilot.diwicon.com/sc6/api/
  areDetailsClosed: undefined
  collectionLink: undefined
▶ map: function () { [native code] }
▶ model: Object
  name: undefined
▶ sourceResource: Object
```

pod full model.contentList[4] === collection/modules/content

The.viewmodel of the content that had been added to the contentList of the pod model by the pod mapper using the content resource's type.

2 142 824 m³
Yearly Profile



STATUS

21 minutes ago
3.695 V
Onboard Char...

21 minutes ago
27.89 °C
Onboard Temp...

21 minutes ago
3.525 V
Onboard Voltage

21 minutes ago
100.0 %
Solar Power

COMMUNICATION

21 minutes ago
-67 dBm
Signal Quality

ALL EVENTS 73

Command Acknowledged by ...
Termoplín POD 01

Command Sent to Device
Termoplín POD 01
Tamás Varga

Quantity modified
Termoplín POD 01
Tamás Varga

Show all..

content module

```
▼ content: Object
  __moduleId__: "contexts/api/resourceTypes/collection"
▶ activate: function () {
  apiLink: "https://pilot.diwicon.com/sc6/api/collections/termoplín-pod-01"
  areDetailsClosed: undefined
  collectionLink: undefined
▶ map: function () { [native code] }
▶ model: Object
  name: undefined
▶ sourceResource: Object
```

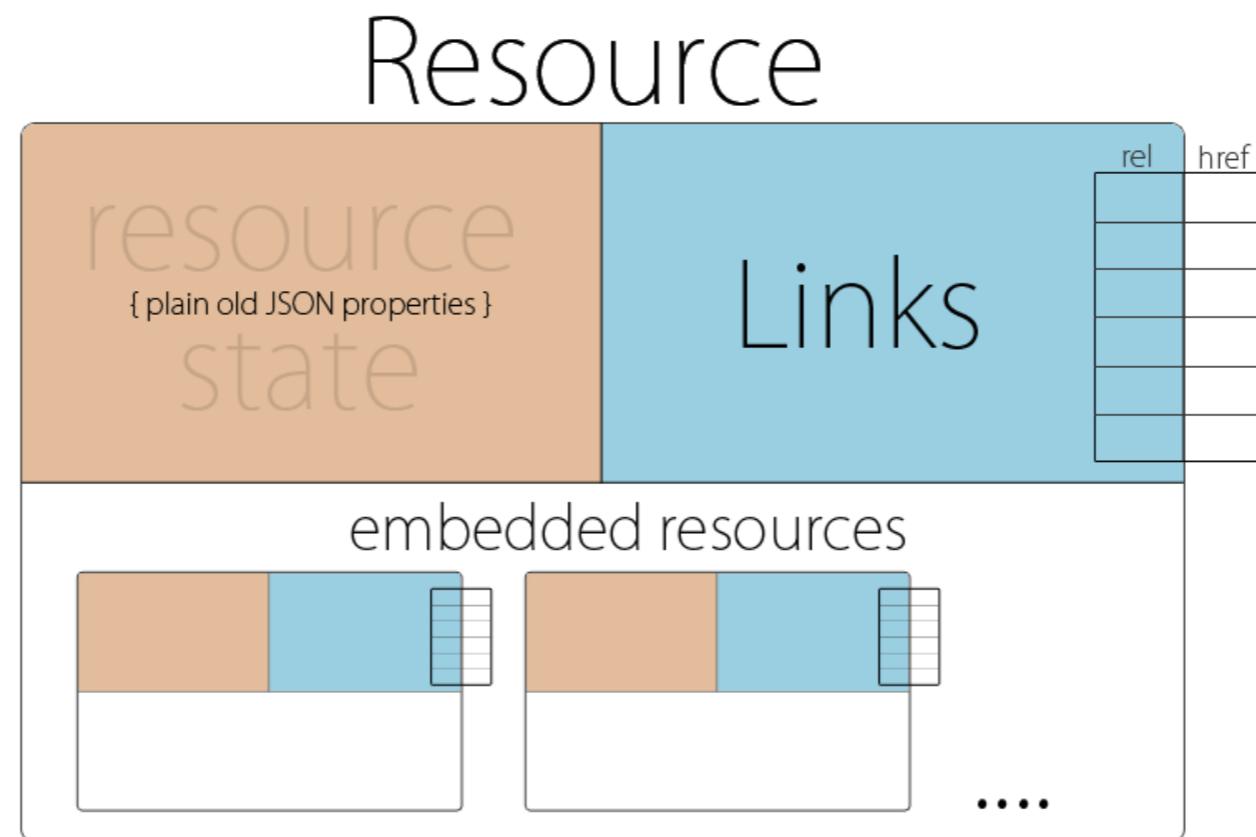
pod full model.contentList[4] === collection/modules/content

The viewmodel of the content that had been added to the contentList of the pod model by the pod mapper using the content resource's type.

Other remarks

API Resource format

- inspiration: HAL - Hypertext Application Language



Build

- Before deploying the web client it is need to be built into one js file. Otherwise all modules are downloaded separately and cause high http traffic.
- How to build a Durandal application: <http://durandaljs.com/documentation/Weyland.html>