

Fixing bugs result

AA03

Juni 2020

# 

# 

# Index

[**Index**](#_7yt3jspbvqmb) **1**

[**Introduction**](#_m2c6rtjq4tcs) **3**

[**Check-box column will be first column in address grid**](#_iv2abpfsno29) **3**

[Description](#_ft8d2xex6zgo) 3

[Solution](#_ihh4c12vbek3) 4

[The old code](#_g1n577a4tekj) 4

[scripts/script.js](#_jp0659v3s49n) 4

[New code](#_v4gcwxjzxkxf) 4

[scripts/script.js](#_o7v7z42nec0o) 4

[Result](#_6tn5wf9ttx9m) 5

[Communication project manager](#_j22y6ulpi16z) 5

[**By default, all check-box will be checked**](#_rggkkwf5ci9w) **6**

[Description](#_ufjx2bige65q) 6

[Solution](#_9bxfmi2suxa5) 6

[The old code](#_rvc247iui2kn) 6

[scripts/script.js](#_80mrh4vnjtcn) 6

[New code](#_fzh3jt8uxi3h) 8

[scripts/script.js](#_v7gi9xbhk3sv) 8

[Result](#_23umvngqd6wn) 10

[Communication project manager](#_ytu6ctdi6x52) 10

[**Rename ‘zoom’ to apply**](#_vq1s2w5b66pa) **11**

[Description](#_uggnlwbh2ucw) 11

[Solution](#_lctl53heplri) 11

[The old code](#_l31r24kcf5h4) 11

[scripts/script.js](#_uscjfgfyimlv) 11

[New code](#_286xayqbppuz) 12

[scripts/script.js](#_s59vasp9abh2) 12

[Result](#_x04jilj52xvb) 14

[Communication project manager](#_w7dutf3ycv9h) 14

[**Place button ‘reload’**](#_ne5sxp7uf4vv) **15**

[Description](#_g9ukvd1vtur4) 15

[Solution](#_fjjo8botzjbf) 15

[The old code](#_vpwyq2hkqpw2) 15

[scripts/script.js](#_3mov1p2k3r04) 15

[The new code](#_h8pu8cctj0mk) 16

[scripts/script.js](#_i8k3alhs2dvr) 17

[Result](#_uacdqqis4vqy) 19

[Communication project manager](#_4vutgqsp5tcx) 19

# 

# 

# Introduction

This bugfix report contains information about the bugs I solved in the project AA03.

This document describes what the bug entailed with a short description and photos. Then it is described how I solved the bugs with a code block.

I have had to look up 4 bugs and fix all 4 in the AA03 application. The application is made in c # and uses leaflet.

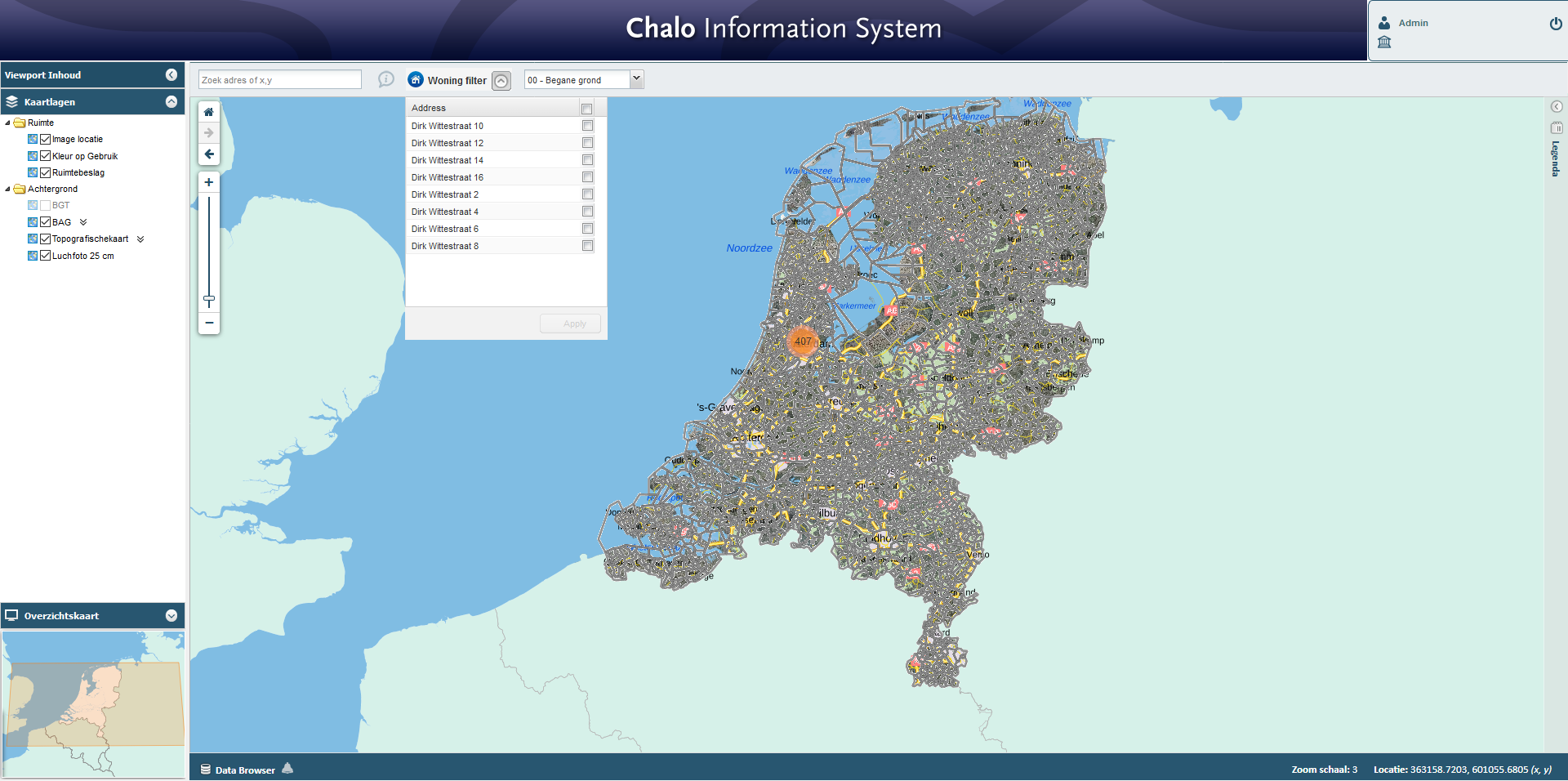
# 

# 

# Check-box column will be first column in address grid

## 

## Description

The address text is placed first then the checkbox is placed. The issue is that the checkbox needs to be placed first. So when you activate the dropdown the checkboxes will be on the left and the addresses will be next to the checkboxes. ****

## Solution

First load checkboxes. After that load the address text. That way the checkboxes will be placed before the address text.

### The old code

#### scripts/script.js

|  |
| --- |
| var myColumnModel = new Ext.grid.ColumnModel([  { id: 'address', header: "Address", width: 210, sortable: true, dataIndex: 'address', },  new Ext.grid.CheckboxSelectionModel(), ]); |

### New code

#### scripts/script.js

|  |
| --- |
| var myColumnModel = new Ext.grid.ColumnModel([  new Ext.grid.CheckboxSelectionModel(),  { id: 'address', header: "Address", width: 210, sortable: true, dataIndex: 'address', }, ]); |

#### 

### Result

With the new code from above this is the result. As you can see the checkboxes are now loaded before the address text.

## Communication project manager

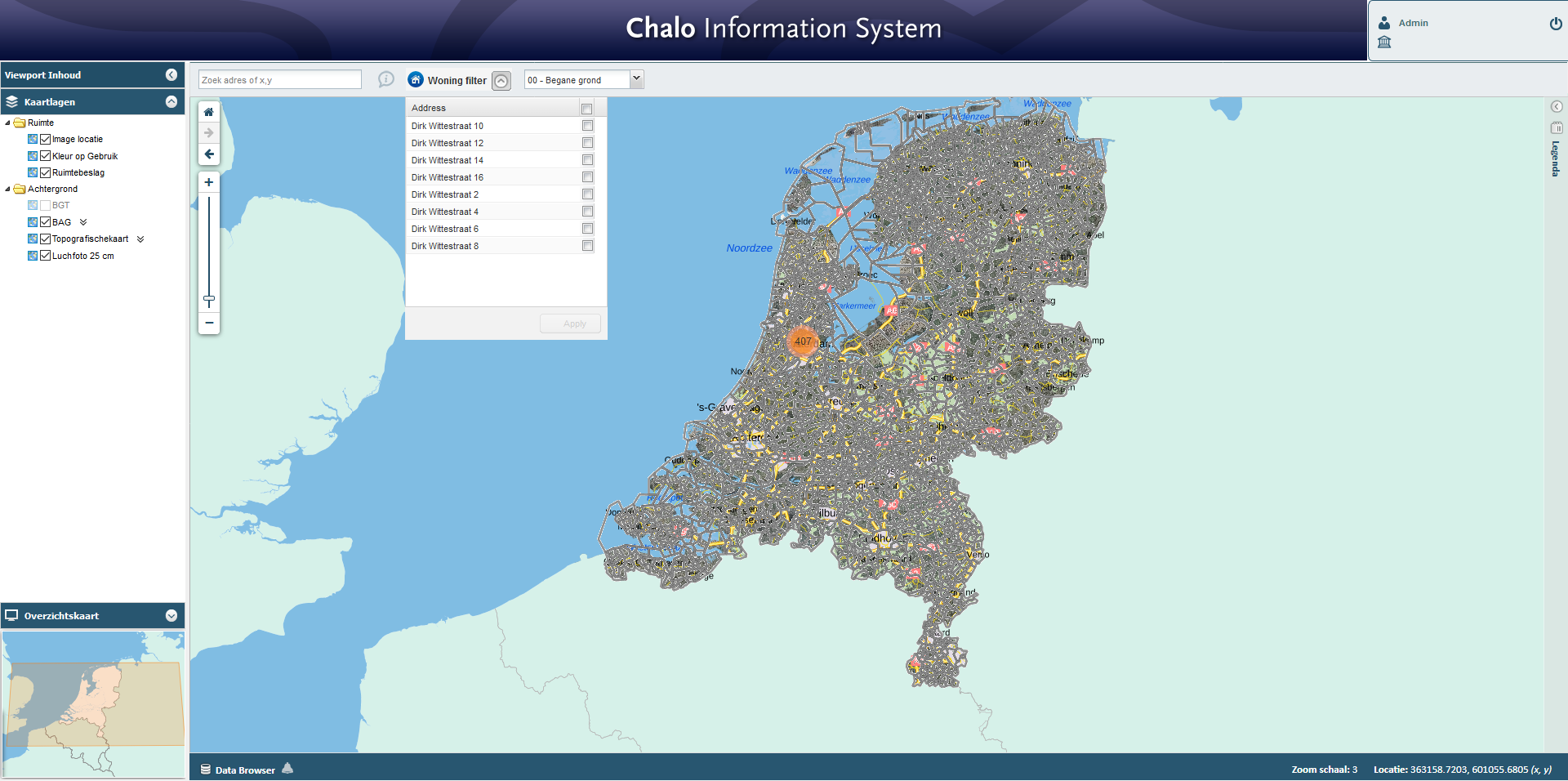
First I thought that the whole dropdown had to move to the first position of the bar. But after communicating with the project manager the real issue came clear.

Problem presented to the client with an appropriate solution. The client agreed with the solution and was therefore implemented.

# 

# By default, all check-box will be checked

## Description

The dropdown shows checkboxes. The issue is that the checkboxes are not selected by default. The intention is that all checkboxes are selected by default so user only needs to press ‘apply’.  


## Solution

Find the function that triggers the checkboxes and make it trigger by default.

### 

### The old code

#### scripts/script.js

|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [  {  text: 'Reload',  id: 'Reloadbtn',  disabled: false,  handler: function () {  GetBuildingInfo();  }  },  {  text: 'Apply',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  },  ]  });  grid.render('mapToolbarPanel'); } |

### New code

Added function set time out with checkall function inside it. So when the grid is rendered the function “selectAll” will be executed and will check all checkboxes by default.

#### scripts/script.js

|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [  {  text: 'Reload',  id: 'Reloadbtn',  disabled: false,  handler: function () {  GetBuildingInfo();  }  },  {  text: 'Apply',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  },  ]  });  grid.render('mapToolbarPanel');    setTimeout(function () { grid.getSelectionModel().selectAll() }, 50); } |

### Result

When the grid is rendered the function “selectAll” will be executed and will check all checkboxes by default.

## 

## Communication project manager

Problem presented to the client with an appropriate solution. The client agreed with the solution and was therefore implemented.

# 

# Rename ‘zoom’ to apply

## Description

The issue is that the button ‘zoom’ needs a different name. The name is unclear and weird right now. The button name will be changed to ‘apply’

## Solution

The name ‘zoom’ is given inside the scripts/script.js file. Simply replace the text ‘zoom’ with ‘apply’.

### 

### The old code

#### scripts/script.js

|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [{  text: 'Zoom',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  }],   }); |

### New code

#### scripts/script.js

|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [{  text: 'Apply',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  }],   }); |

### Result

The ‘zoom’ button is now the ‘apply’ button.



## Communication project manager

Problem presented to the client with an appropriate solution. The client agreed with the solution and was therefore implemented.

# 

# Place button ‘reload’

## Description

The issue is that there is no ‘reload’ button inside the address dropdown. The ‘reload’ button will be located before the ‘apply’ button.



## Solution

Add a reload button next to the apply button.

### 

### The old code

#### scripts/script.js

|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [  {  {  text: 'Apply',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  },  ],  }); |

### The new code

Inside the “buttons” I added the button reload.

#### scripts/script.js

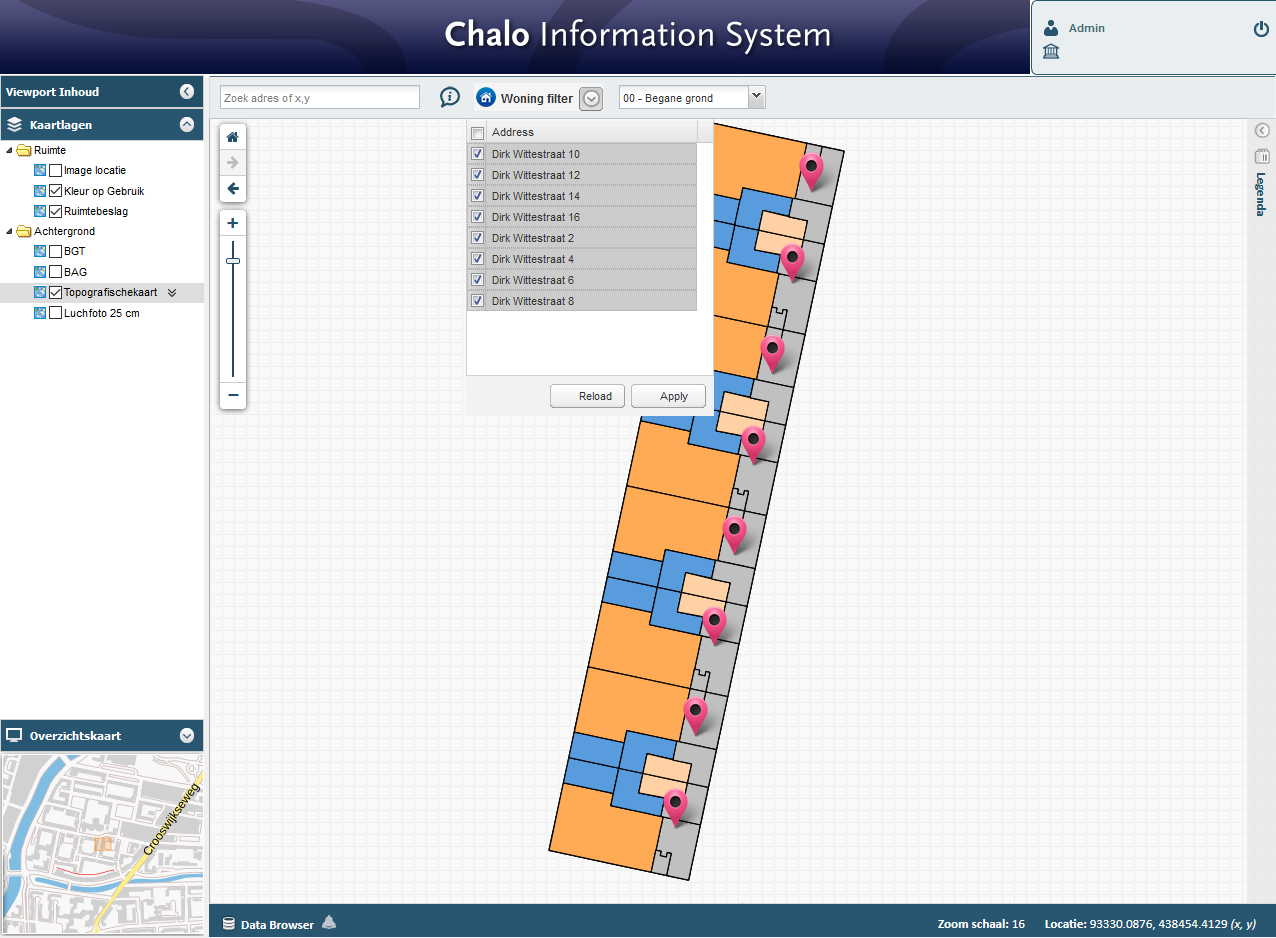
|  |
| --- |
| var grid = new Ext.grid.GridPanel({  id: "addressGrid",  width: 250,  height: 300,  scrollable: true,  store: dataStore,  stripeRows: true,  cm: myColumnModel,  sm: mySelectionModel,  border: true,  layout: 'fit',  buttons: [  {  text: 'Reload',  id: 'Reloadbtn',  disabled: false,  handler: function () {  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  },  {  text: 'Apply',  id: 'zoombtn',  disabled: true,  handler: function () {  Ext.getCmp('addgridbtn').setIconCls("expandIcon")  Ext.getCmp('addgridbtn').toggle(false);  Ext.getCmp('addressGrid').hide();  var selectedValues = Ext.getCmp('addressGrid').getSelectionModel();  var buildingAddresses = [];  for (var i = 0; i < selectedValues.selections.items.length; i++) {  buildingAddresses.push(selectedValues.selections.items[i].data.address);  }  var zoomBoundingbox = GetSelectedBuildingInfo(buildingAddresses);  ZoomToSelectedBuilding(zoomBoundingbox);  RenderAddressPinPoints(buildingAddresses);  }  },  ]  });  grid.render('mapToolbarPanel');    setTimeout(function () { grid.getSelectionModel().selectAll() }, 50); } |

### 

### 

### Result

When the button ‘reload’ is pressed all the addresses will be loaded again.



## Communication project manager

Problem presented to the client with an appropriate solution. The client agreed with the solution and was therefore implemented.

### 