



Hands-On with Mendix 7

March 30 2017

Andrej Koelewijn, Bart Luijten, Jan de Vries

Mendix 7

- ▶ Smart Apps - Make your apps intelligent, proactive, and contextual
- ▶ Mendix Connector Kit - Defining integrations is now a breeze
- ▶ Mendix Web Modeler - Experience the value of collaborative development
- ▶ Native Mobile Experience – Anywhere, Always On
- ▶ Mendix Application Test Suite - Embed testing in your application life-cycle
- ▶ Mendix Application Performance Monitor - Get a grip on performance early on

Mendix 7

- ▶ Resilient and scalable Apps – Stateless Mendix Runtime
- ▶ Smart Apps - Make your apps intelligent, proactive and contextual
- ▶ Mendix Connector Kit - Defining integrations is now a breeze
- ▶ Mendix Web Modeler - Experience the value of collaborative development
- ▶ Native Mobile Experience - Anywhere. Always On
- ▶ Mendix Application Test Suite - Embed testing in your application lifecycle
- ▶ Mendix Application Performance Monitor - Get a grip on performance from early on

Mendix 7

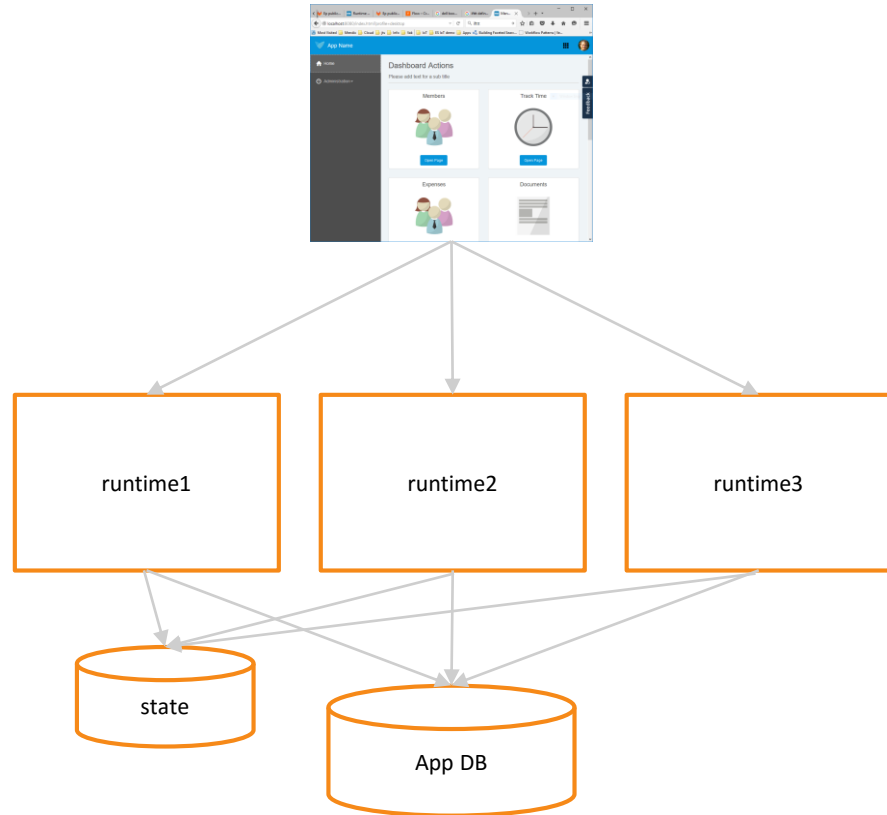
- ▶ Resilient and scalable Apps – Stateless Mendix Runtime
 - Why a stateless runtime?
 - How does it work?
 - Best practices
- ▶ Migration to Mendix 7

Mendix 7

- ▶ Resilient and scalable Apps – Stateless Mendix Runtime
 - Why a stateless runtime?
 - How does it work?
 - Best practices
- ▶ Migration to Mendix 7
- ▶ REST consume

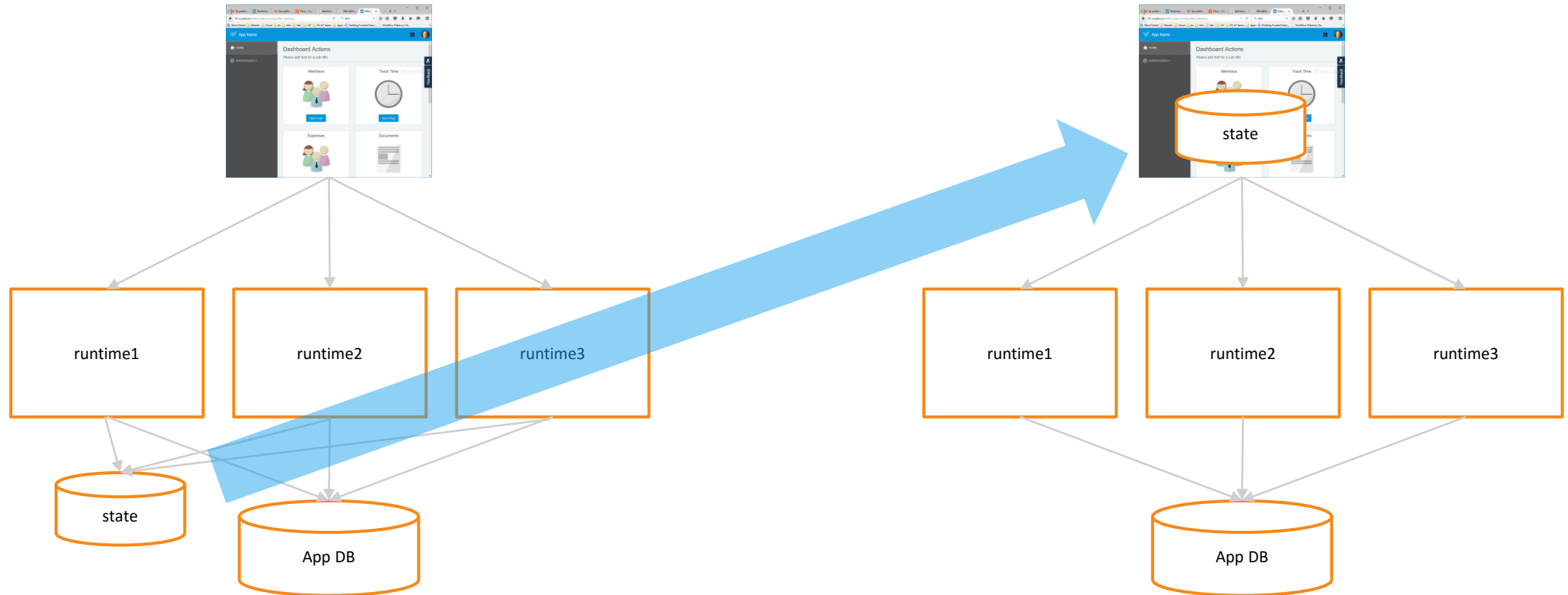
Why a Stateless Runtime?

Mendix 6 – Shared State Database



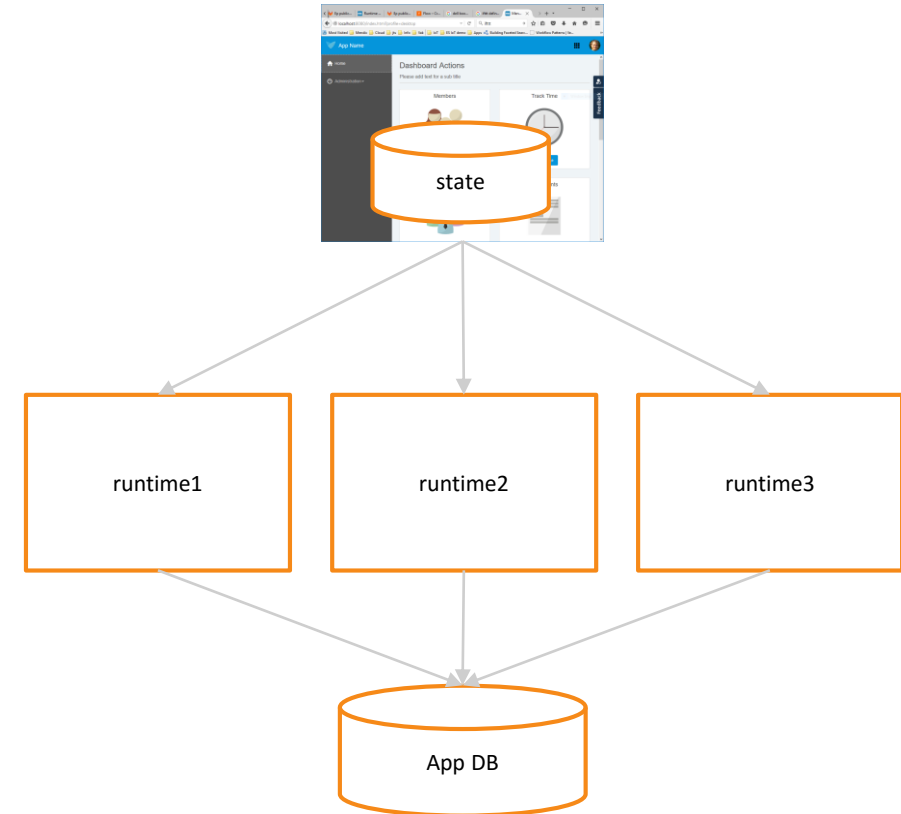
- Runtime state shared in a state database: Redis
- Pros
 - Transparent to client
- Cons
 - Does not scale to large amounts of objects in session
 - Complex garbage collection
 - Complex infrastructure

Mendix 7 - Stateless runtime



Mendix 7 - Stateless Runtime

- ▶ State is stored in client:
 - Objects in JavaScript memory
 - Garbage collected as soon as possible
 - Protected against changes of read-only data
 - Server communication optimized:
 - ▶ Minimal roundtrips
 - ▶ Minimal dataset needed by server
 - Model analysis to ensure server requests are as small as possible

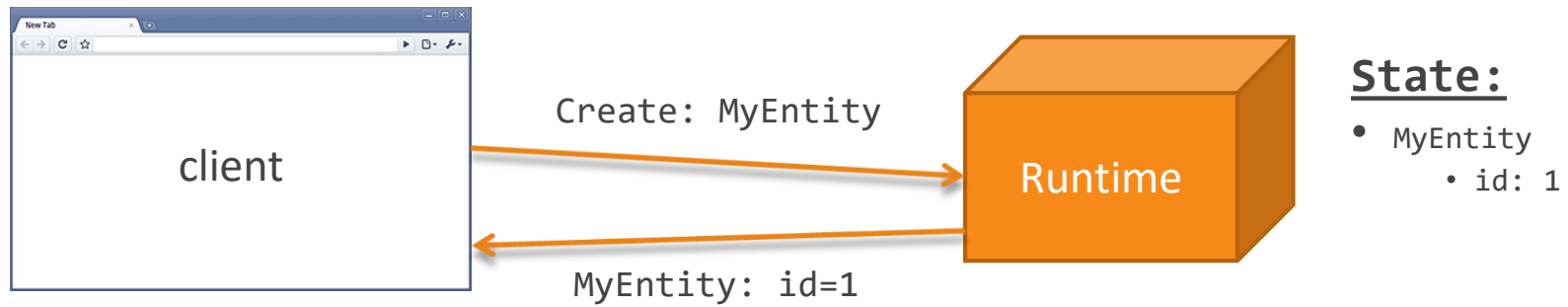


How Does This Work?

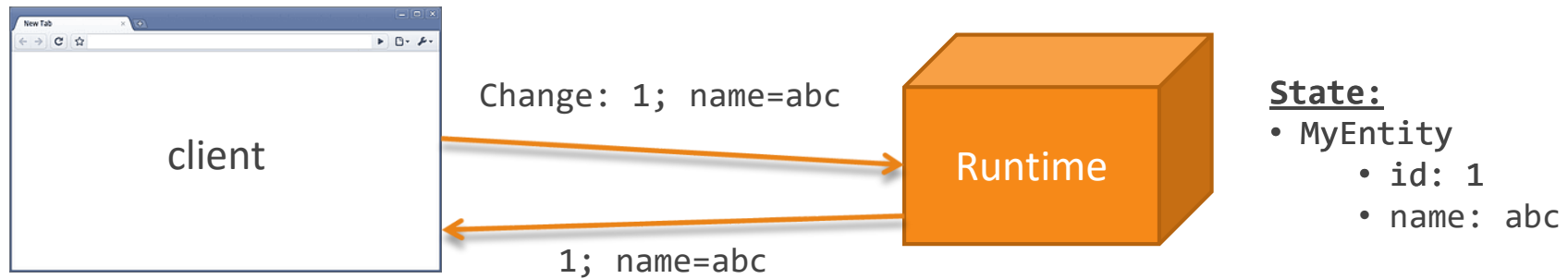
Important State Concepts

- ▶ What will be kept in state by client?
- ▶ What will be included in server communication?
- ▶ When to create association with session?

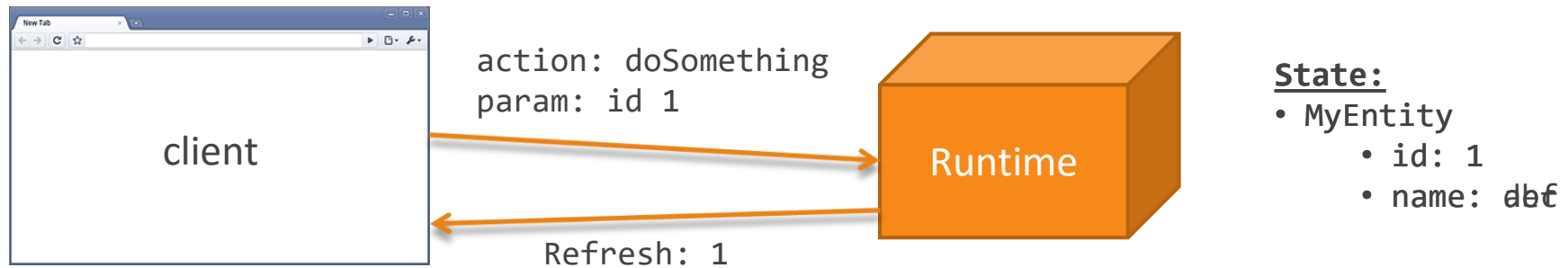
State in Mendix 6



State in Mendix 6



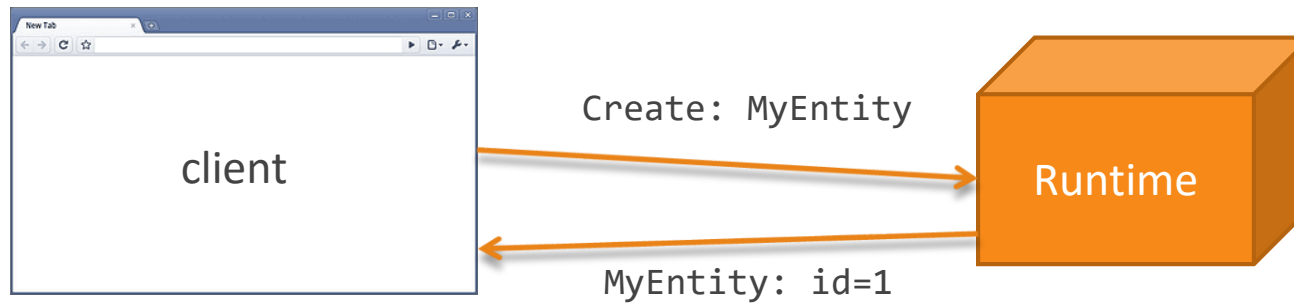
State in Mendix 6



State in Mendix 6

- ▶ Session state stored server side
- ▶ Downsides
 - Scaling out (state synchronization)
 - Garbage collection
 - Chatty protocol
 - Runtime Memory usage

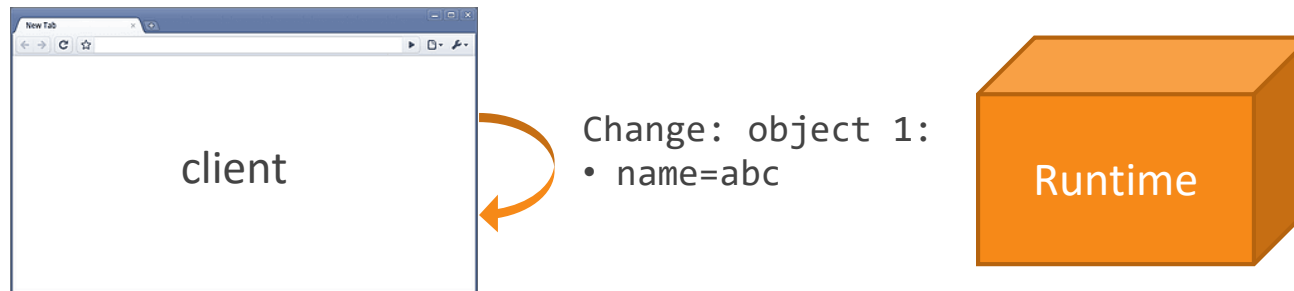
State in Mendix 7



State:

- MyEntity
 - id: 1

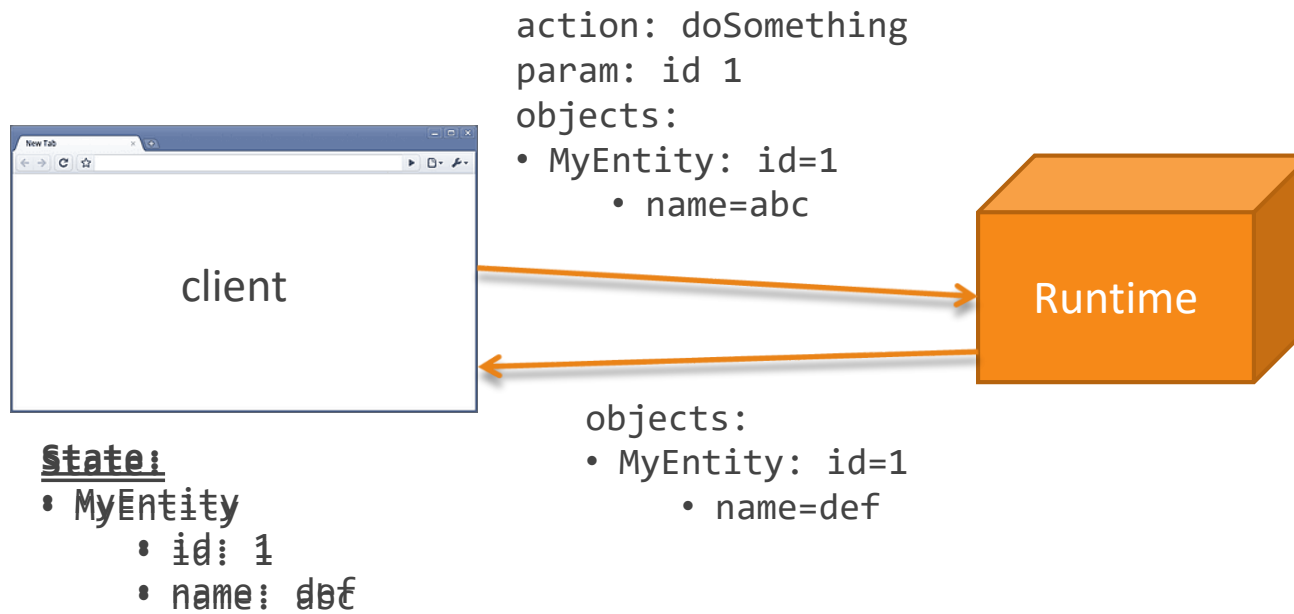
State in Mendix 7



State:

- MyEntity
 - id: 1
 - name: abc

State in Mendix 7



State in Mendix 7

- ▶ All user session state stored client side
- ▶ Benefits
 - Fewer limitations on server scaling
 - Lower memory requirements in Runtime
 - Fewer roundtrips
 - More efficient garbage collection
 - Better insight for developers
- ▶ Potential side effects
 - Larger requests and responses
 - Due to optimizations in Mendix some apps actually have smaller requests!

Mendix 7: Impact on Your Projects

- ▶ Request input
 - What client state needs to be included when calling the server
- ▶ Reachable network
 - What client state can be accessed from pages

Mendix 7: Impact on Your Projects

- ▶ Request input calculated based on the reachable network of:
 - Request parameters (like microflow inputs and associations used in a microflow)
 - ▶ Optimizations are disabled for java actions and service calls
 - Current User object
 - Current Session object
- ▶ Reachable network is calculated based on objects available on the client
 - Garbage collection (GC) limits the reachable network
 - Static analysis of the model is used to determine server side data usage
 - GC keeps all reachable ...
 - ▶ ... *NPE objects* from current user, session and subscribed objects
 - ▶ ... *changed objects* from current user, session and subscribed objects

Best Practices

Best practice

- ▶ Minimize the number of in-use objects in your session
- ▶ Commit or roll back all changes to persistable objects before the end of the main microflow
- ▶ Link non-persistable objects that have long life spans to the current Session object
- ▶ Map only those parts of a web service integration that are necessary
- ▶ Delete any non-persistable objects as soon as they are no longer necessary
- ▶ Don't use non-persistable objects in layouts



Docs > General How-To's > Best Practices > Best Practices for App Performance

Search documentation

Jump to:

- > Mendix Cloud
- > Tips & Tricks
- ▼ **Best Practices**
 - Best Practices for App Performance**
 - Best Practices for Component Based...
 - Security Best Practices and Improve...
 - Mendix Development Best Practices
 - How to convert your application fro...
 - How to execute an SQL statement o...
 - Inheritance vs. 1-1 association
 - SIG - Mendix performance subjects ...

Best Practices for App Performance in Mendix 7

Edit on Github

Last update: 10 days ago

Table of contents

- Overview
- Limiting the Number of Objects
- Changed Objects
- Workflow Objects
- Integrations

Feedback

<https://docs.mendix.com/howtogeneral/bestpractices/best-practices-for-app-performance-in-mendix-7>

Overview

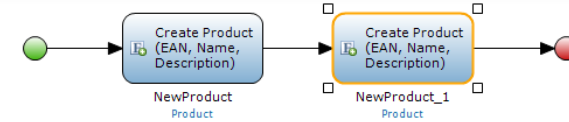
Minimize In-Use Objects in Your Session

- ▶ In-use (dirty) objects cannot be garbage collected
 - Will increase memory usage of your client
 - May increase request size when calling server
- ▶ Consider
 - NPEs pointing to many objects used in layout
 - ▶ prevents GC as they stay reachable from subscribed object in layout
 - NPEs pointing to current user object and current session
 - ▶ they need to be manually deleted when no longer necessary

Minimize Dirty State at End of Microflow

- ▶ Dirty state (new or changed persistent objects) needs to be held by the browser until it's saved to the database
- ▶ Commit or roll back the changes to persistable objects before the end of the main microflow

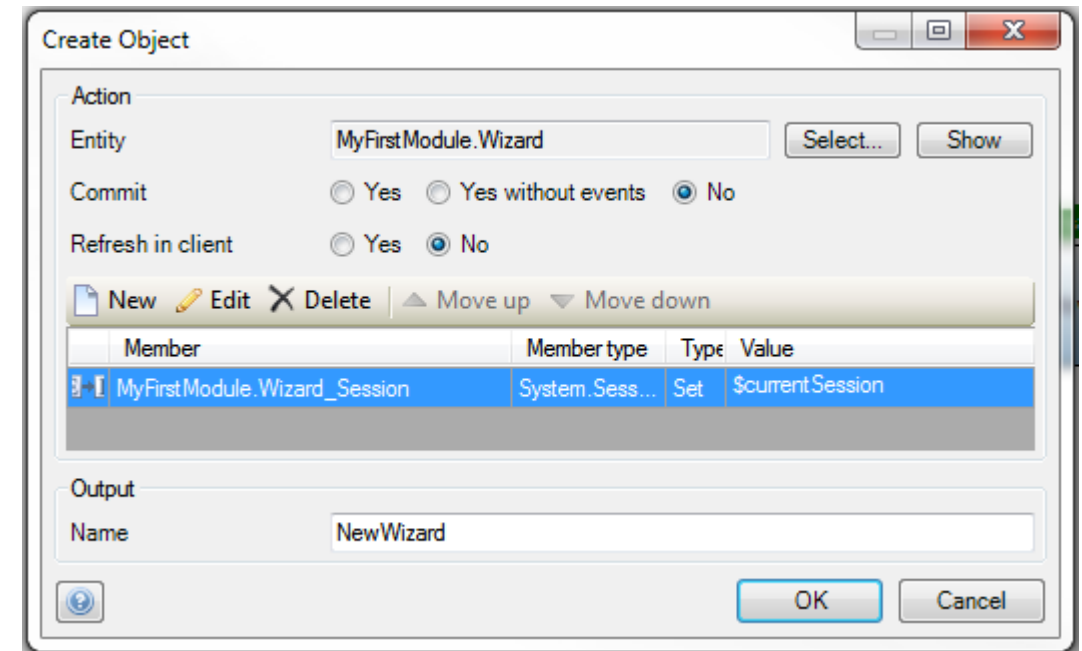
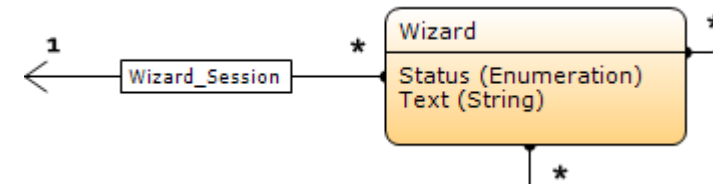
Member	Member type	Type	Value
EAN	String (13)	Set	'124'
Name	String (200)	Set	'Chair'
Description	String (200)	Set	'A desk chair'



```
Preview
{, ...}
  actionResult: null
  changes: {8444249301319781: {EAN: {value: "123"}, Description: {value: "A desk"}, Name: {value: "Table"}}, ...}
    8444249301319781: {EAN: {value: "123"}, Description: {value: "A desk"}, Name: {value: "Table"}}
      Description: {value: "A desk"}
      EAN: {value: "123"}
      Name: {value: "Table"}
    8444249301319782: {EAN: {value: "124"}, Description: {value: "A desk chair"}, Name: {value: "Chair"}}
      Description: {value: "A desk chair"}
      EAN: {value: "124"}
      Name: {value: "Chair"}
  newpersistable: ["8444249301319781", "8444249301319782"]
    0: "8444249301319781"
    1: "8444249301319782"
  objects: [{objectType: "Main.Product", guid: "8444249301319781", ...}, ...]
```

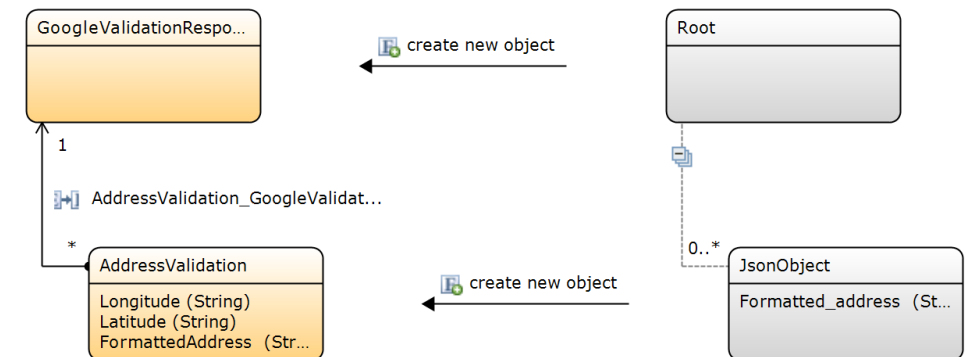
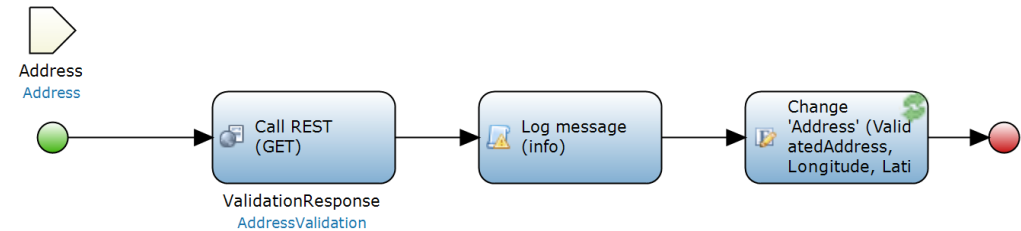
Link Long-Lived NPEs to Current Session

- ▶ By linking to `$currentSession`
 - Garbage collection knows it cannot be garbage collected
 - You can easily retrieve it
- ▶ Delete the NPE when no longer used



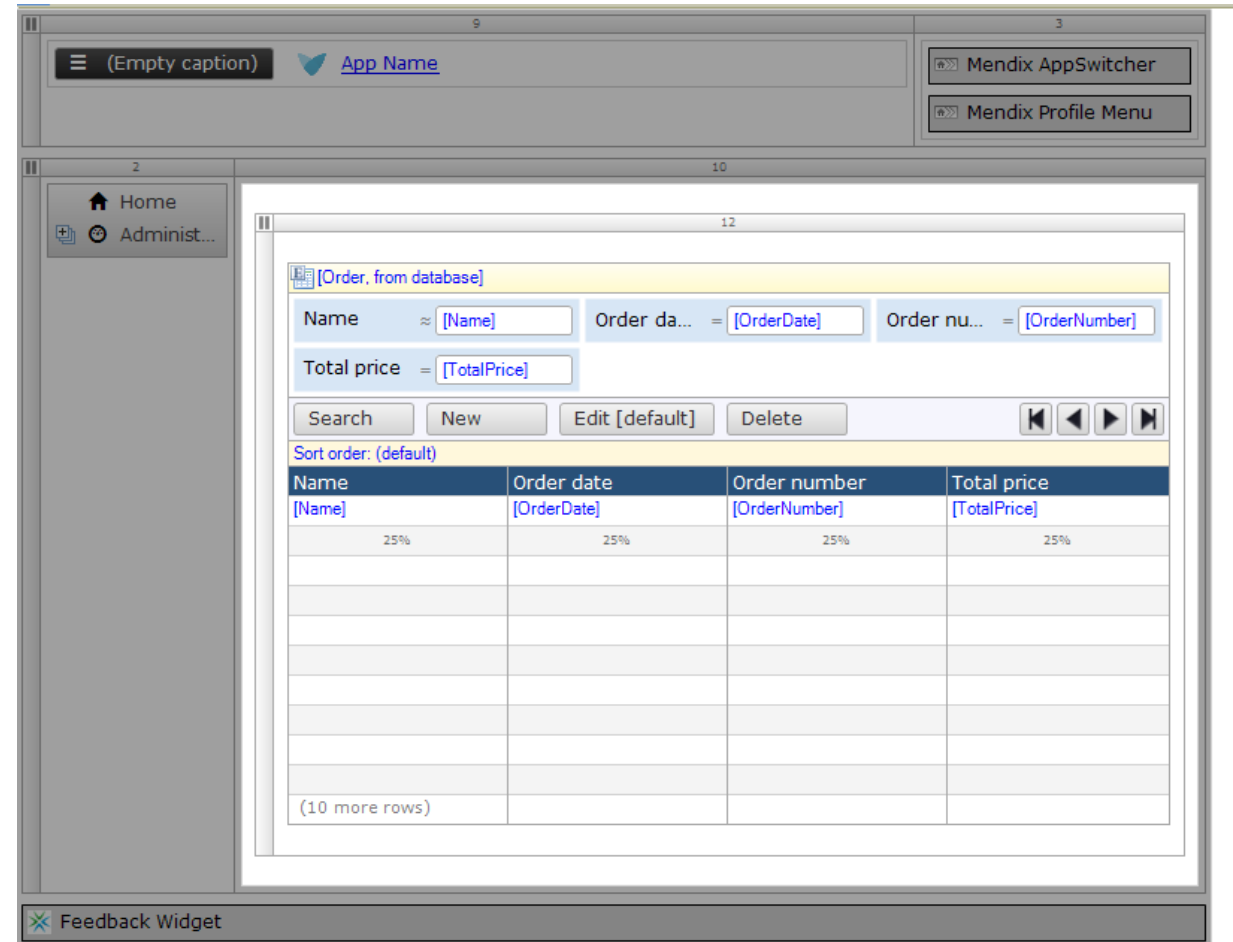
Integration: Calling Services

- ▶ NPEs resulting from a service call will all be sent to the browser
- ▶ Tips
 - Map only those parts of a web service integration that are necessary
 - Delete NPEs as soon as they are no longer necessary
 - Use database (persistent entities) to cache service call results



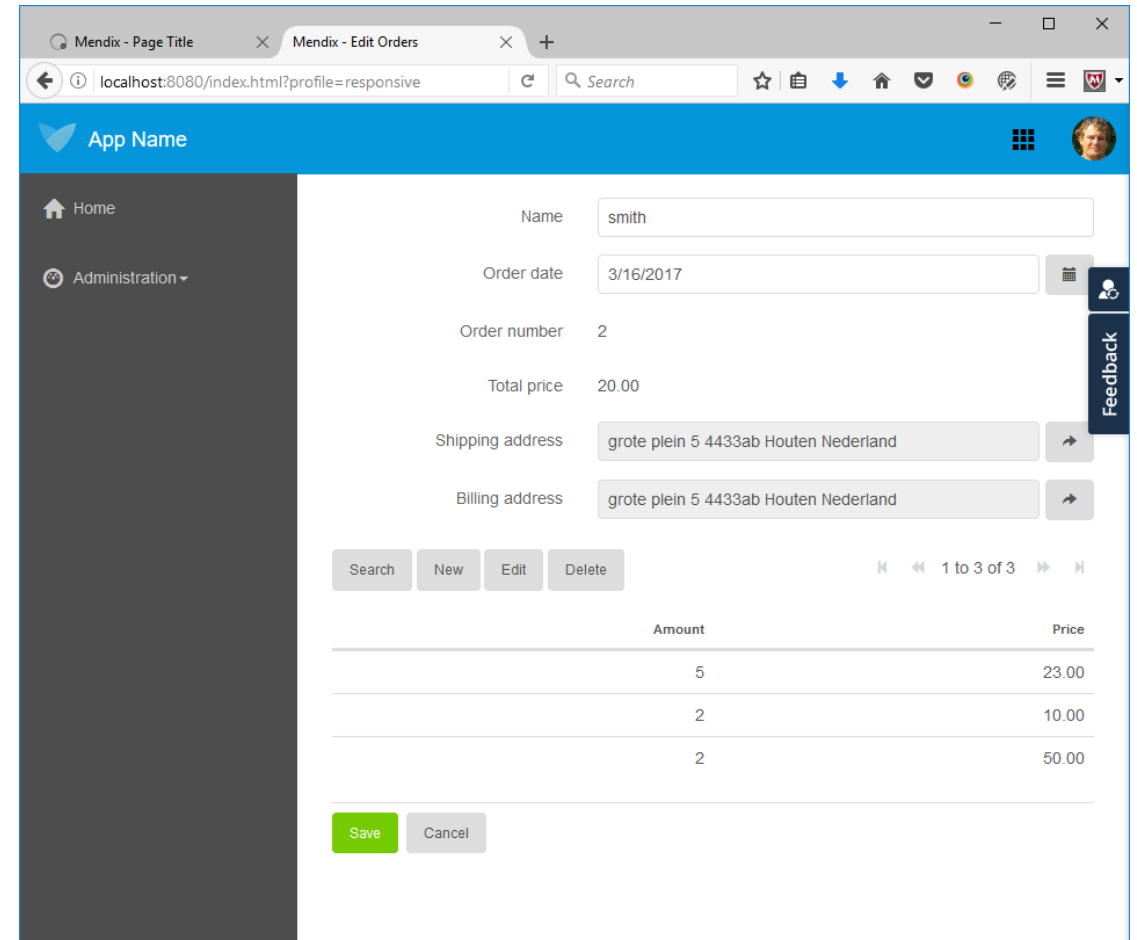
Don't Use NPEs in Layouts

- ▶ Objects in layouts can be on the screen for a long time
- ▶ These will be sent back and forth between the client and Runtime very often



Browser State

- ▶ Browser refresh loses any unsaved changes
- ▶ Multiple browser tabs behave like separate browsers
 - Each browser tab has own client state



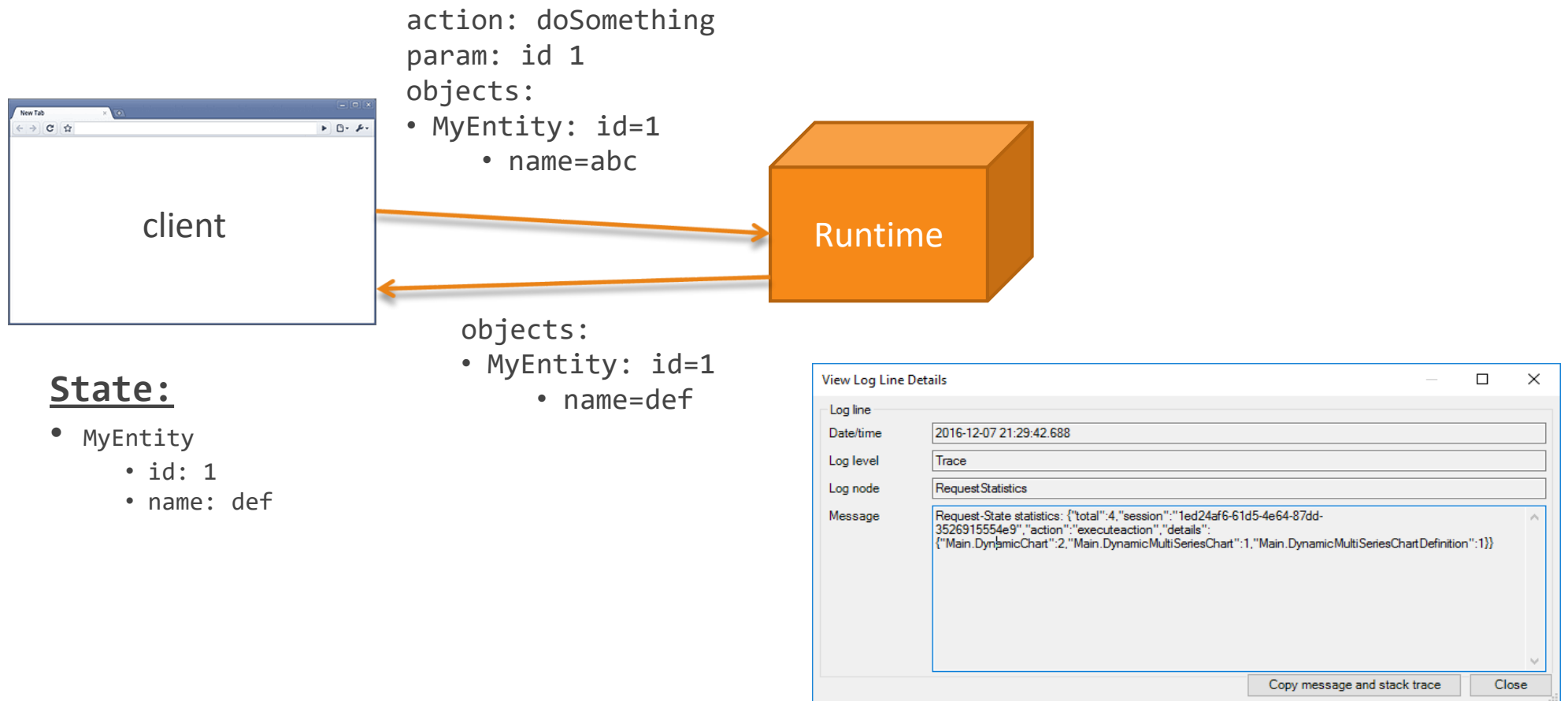
Custom Widgets: Use widget.subscribe

- ▶ Objects with subscribed widgets will not be garbage collected
 - widget.subscribe will automatically unsubscribe if no longer used
 - data.subscribe needs manual data.unsubscribe
- ▶ More info:
 - <https://apidocs.mendix.com/7/client/mx.data.html>
 - https://apidocs.mendix.com/7/client/mxui_widget_WidgetBase.html

```
▼ Object 1
  ▼ Main.Address: Object
    ▼ 8162774324609026: Object
      ▼ subscribedWidgets: Array[2]
        0: "Main.Orders_NewEdit.referenceSelector1"
        1: "Main.Orders_NewEdit.referenceSelector2"
        length: 2
      ► __proto__: Array[0]
      ► __proto__: Object
      ► __proto__: Object
    ▼ Main.Order: Object
      ▼ 7881299347898370: Object
        ▼ subscribedWidgets: Array[8]
          0: "Main.Orders_NewEdit.dataView1"
          1: "Main.Orders_NewEdit.textBox6"
          2: "Main.Orders_NewEdit.datePicker2"
          3: "Main.Orders_NewEdit.textBox7"
          4: "Main.Orders_NewEdit.textBox5"
          5: "Main.Orders_NewEdit.referenceSelector1"
          6: "Main.Orders_NewEdit.referenceSelector2"
          7: "Main.Orders_NewEdit.grid1"
          length: 8
        ► __proto__: Array[0]
        ► __proto__: Object
        ► __proto__: Object
        ► __proto__: Object
```

Developer Support

Monitoring Request Size in Server Log



Browser Developer Tools

Mendix - Edit Orders

localhost:8080/index.html?profile=responsive

mendix

Order date: 2/16/2017

Name: qer qer

Address: er qe

Total amount: 0.00

Search New Edit Delete

1 to 1 of 1

Product name	Quantity	Price
r eg werg	3	23

Save Cancel

Feedback

Browser Developer Tools

Mendix - Edit Orders

localhost:8080/index.html?profile=responsive

mendix

Order date: 2/16/2017

Name: qer qer

Address: er qe

Total amount: 0.00

Search New Edit Delete

Product name	Quantity
er qe	3

Save Cancel

Developer Tools - http://localhost:8080/index.html?profile=responsive

Elements Console Sources Network Timeline Profiles Application Security Audits AngularJS

View: [Icons] Preserve log Disable cache Offline No throttling

Filter [] Regex Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

5000 ms 10000 ms 15000 ms 20000 ms 25000 ms 30000 ms 35000 ms 40000 ms 45000 ms

Name Headers Preview Response Cookies Timing

- xas/
- Orders_NewEdit.page.xml?636227640449621661
- AppCloudMasterLayout.layout.xml?636227640449621661
- xas/
- xas/
- Homepage.page.xml?636227640449621661
- Sidebar_Full_Responsive.layout.xml?636227640449621661
- buttonservices.js?PP_6.20
- xas/
- xas/
- Orders_NewEdit.page.xml?636227640449621661
- AppCloudMasterLayout.layout.xml?636227640449621661
- xas/
- xas/

```
{
  "mxobjects": [
    {
      "objectType": "MyFirstModule.Orders",
      "guid": "3377699720527875",
      "attributes": {
        "Address": {
          "value": "er qe",
          "readOnly": true
        },
        "Name": {
          "value": "qer qer"
        },
        "OrderDate": {
          "value": 1487199600000
        },
        "TotalAmount": {
          "value": "0",
          "readOnly": true
        }
      }
    }
  ]
}
```

14 requests | 3.2 KB transferred

Browser Developer Tools

The screenshot shows a web browser window with the URL `localhost:8080/index.html?profile=respons`. The page title is "mendix". The form contains the following fields and buttons:

- Order date: 2/16/201
- Name: qer qer
- Address: er qe
- Total amount: 0.00
- Buttons: Search, New, Edit, Delete
- Table with columns: Product name, Quantity
- Table row: r eg werg, 3
- Buttons: Save, Cancel

```
▼ {resets: {12103423998558612: ["Main.OrderLines_Orders", "Amount", "Main.OrderLines_Products"],...},...}
  actionResult: null
  ▼ commits: ["12103423998558612", "11258999068426645", "12103423998558613"]
    0: "12103423998558612"
    1: "11258999068426645"
    2: "12103423998558613"
  ▼ instructions: [{target: "system", type: "close", args: {},...}, {target: "system", type: "refresh_object_list",...},...]
    ► 0: {target: "system", type: "close", args: {},...}
    ► 1: {target: "system", type: "refresh_object_list",...}
    ► 2: {target: "system", type: "refresh_class", args: {classnames: ["Main.Order", "Main.OrderLine"],...}}
  ▼ objects: [{objectType: "Main.OrderLine", guid: "12103423998558612",...},...]
    ▼ 0: {objectType: "Main.OrderLine", guid: "12103423998558612",...}
      ▼ attributes: {Main.OrderLines_Orders: {value: "11258999068426645"}, Price: {value: "50", readonly: true},...}
        ▼ Amount: {value: 5}
          value: 5
        ▼ Main.OrderLines_Orders: {value: "11258999068426645"}
          value: "11258999068426645"
        ▼ Main.OrderLines_Products: {value: "11821949021847554"}
          value: "11821949021847554"
        ▼ Price: {value: "50", readonly: true}
          readonly: true
          value: "50"
        guid: "12103423998558612"
        hash: "6Qw6hrN/lJY4fagBDk9XYj30/dqcT1FjN+Qye5wpink="
        objectType: "Main.OrderLine"
      ▼ 1: {objectType: "Main.Order", guid: "11258999068426645",...}
        ▼ attributes: {Main.Orders_BillingAddress: {value: "11540474045136897"}, TotalPrice: {value: "140", readonly: true},...}
          ► Main.Orders_BillingAddress: {value: "11540474045136897"}
          ► Main.Orders_ShippingAddress: {value: "11540474045136897"}
          ► Name: {value: "Rutte"}
          ► OrderDate: {value: 1490652000000}
          ► OrderNumber: {value: 11, readonly: true}
          ► TotalPrice: {value: "140", readonly: true}
          guid: "11258999068426645"
          hash: "eE3QcpbxexZNOkiXTd96iSdNjNrt7WSUZrgiDFk9Ew4g="
          objectType: "Main.Order"
        ▼ 2: {objectType: "Main.OrderLine", guid: "12103423998558613",...}
          ▼ attributes: {Main.OrderLines_Orders: {value: "11258999068426645"}, Price: {value: "90", readonly: true},...}
            ► Amount: {value: 6}
            ► Main.OrderLines_Orders: {value: "11258999068426645"}
            ► Price: {value: "90", readonly: true}
```

The screenshot shows a browser window with a console error. The error message is partially visible and includes the text `75",...}]}` and `175",...}]}`. The error is highlighted in red.

Browser Developer Tools

Developer Tools - <http://localhost:8080/index.html?profile=responsive>

State details per entity (2:13:17 PM): Object

- MyFirstModule.Orders: Object
 - 3377699720527875: Object
 - subscribedWidgets: Array[6]
 - 0: "MyFirstModule.Orders_NewEdit.dataView1"
 - 1: "MyFirstModule.Orders_NewEdit.datePicker2"
 - 2: "MyFirstModule.Orders_NewEdit.textBox6"
 - 3: "MyFirstModule.Orders_NewEdit.textBox7"
 - 4: "MyFirstModule.Orders_NewEdit.textBox5"
 - 5: "MyFirstModule.Orders_NewEdit.grid1"
 - length: 6
 - __proto__: Array[0]
 - __proto__: Object
- __proto__: Object
 - __defineGetter__: __defineGetter__()
 - __defineSetter__: __defineSetter__()
 - __lookupGetter__: __lookupGetter__()
 - __lookupSetter__: __lookupSetter__()
 - constructor: Object()
 - hasOwnProperty: hasOwnProperty()
 - isPrototypeOf: isPrototypeOf()
 - propertyIsEnumerable: propertyIsEnumerable()
 - toLocaleString: toLocaleString()
 - toString: toString()
 - valueOf: valueOf()
 - get __proto__: __proto__()
 - set __proto__: __proto__()
- __proto__: Object
 - __defineGetter__: __defineGetter__()
 - __defineSetter__: __defineSetter__()
 - __lookupGetter__: __lookupGetter__()
 - __lookupSetter__: __lookupSetter__()
 - constructor: Object()
 - hasOwnProperty: hasOwnProperty()
 - isPrototypeOf: isPrototypeOf()
 - propertyIsEnumerable: propertyIsEnumerable()
 - toLocaleString: toLocaleString()
 - toString: toString()
 - valueOf: valueOf()
 - get __proto__: __proto__()
 - set __proto__: __proto__()

Ctrl + Alt + G

More client object state info:

- * What objects are in client state?
- * What widgets are using these objects?

14 requests | 3.2 KB transferred

Browser developer tools

The image shows a Mendix application interface on the left and browser developer tools on the right. The application has a blue header with the Mendix logo and a form with fields for 'Order date', 'Name', 'Address', and 'Total amount'. Below the form are buttons for 'Search', 'New', 'Edit', and 'Delete', and a 'Save' button. The developer tools show the 'Console' tab with 'State details per entity' logs. The logs show the state of the application at different times, with the state being updated as the user interacts with the form. The state is an object with properties like 'MyFirstModule.Orders' and 'MyFirstModule.OrderLine'. The state is updated as the user interacts with the form, and the state is kept in the client state depending on the needs of the page and modified state.

Developer Tools - http://localhost:8080/index.html?profile=responsive

State details per entity (2:13:17 PM):

```
Object {
  MyFirstModule.Orders: Object
    3377699720527875: Object
      subscribedWidgets: Array[6]
        0: "MyFirstModule.Orders_NewEdit.dataView1"
        1: "MyFirstModule.Orders_NewEdit.datePicker2"
        2: "MyFirstModule.Orders_NewEdit.textBox6"
        3: "MyFirstModule.Orders_NewEdit.textBox7"
        4: "MyFirstModule.Orders_NewEdit.textBox5"
        5: "MyFirstModule.Orders_NewEdit.grid1"
      length: 6
      __proto__: Array[0]
    __proto__: Object
  __proto__: Object
}
```

State details per entity (2:22:04 PM):

```
Object {
  MyFirstModule.Orders: Object
    3377699720527875: Object
      subscribedWidgets: Array[6]
      __proto__: Object
  __proto__: Object
}
```

State details per entity (2:22:23 PM):

```
Object {
  MyFirstModule.OrderLine: Object
    2533274790395908: Object
      subscribedWidgets: Array[4]
        0: "MyFirstModule.OrderLine_NewEdit.dataView1"
        1: "MyFirstModule.OrderLine_NewEdit.textBox6"
        2: "MyFirstModule.OrderLine_NewEdit.textBox7"
        3: "MyFirstModule.OrderLine_NewEdit.textBox5"
      length: 4
      __proto__: Array[0]
    __proto__: Object
  __proto__: Object
}
```

State details per entity (2:22:42 PM):

```
Object {
  MyFirstModule.OrderLine: Object
    2533274790395908: Object
      referencedBy: Object
      subscribedWidgets: Array[4]
      __proto__: Object
  __proto__: Object
  MyFirstModule.Orders: Object
    3377699720527875 (changed): Object
      changes: Object
      referencedBy: Object
      __proto__: Object
  __proto__: Object
}
```

Objects kept in client state depends on needs of page and modified state

Migrating to Mendix 7

Migration

- ▶ Make sure you have a backup!

Migration

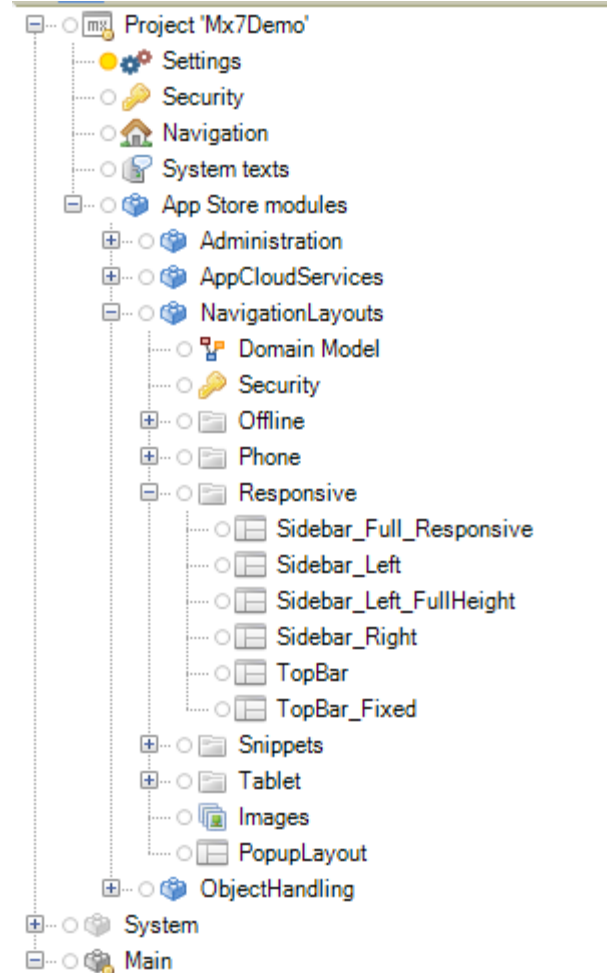
- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library

Migration

- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library

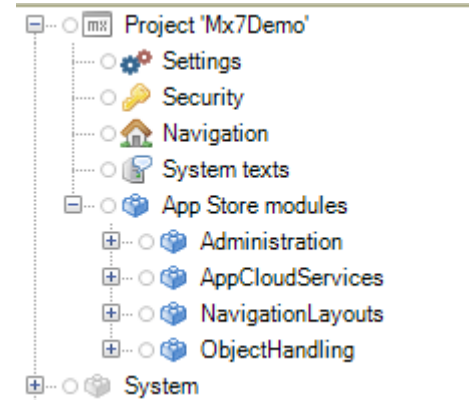
Migration

- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library



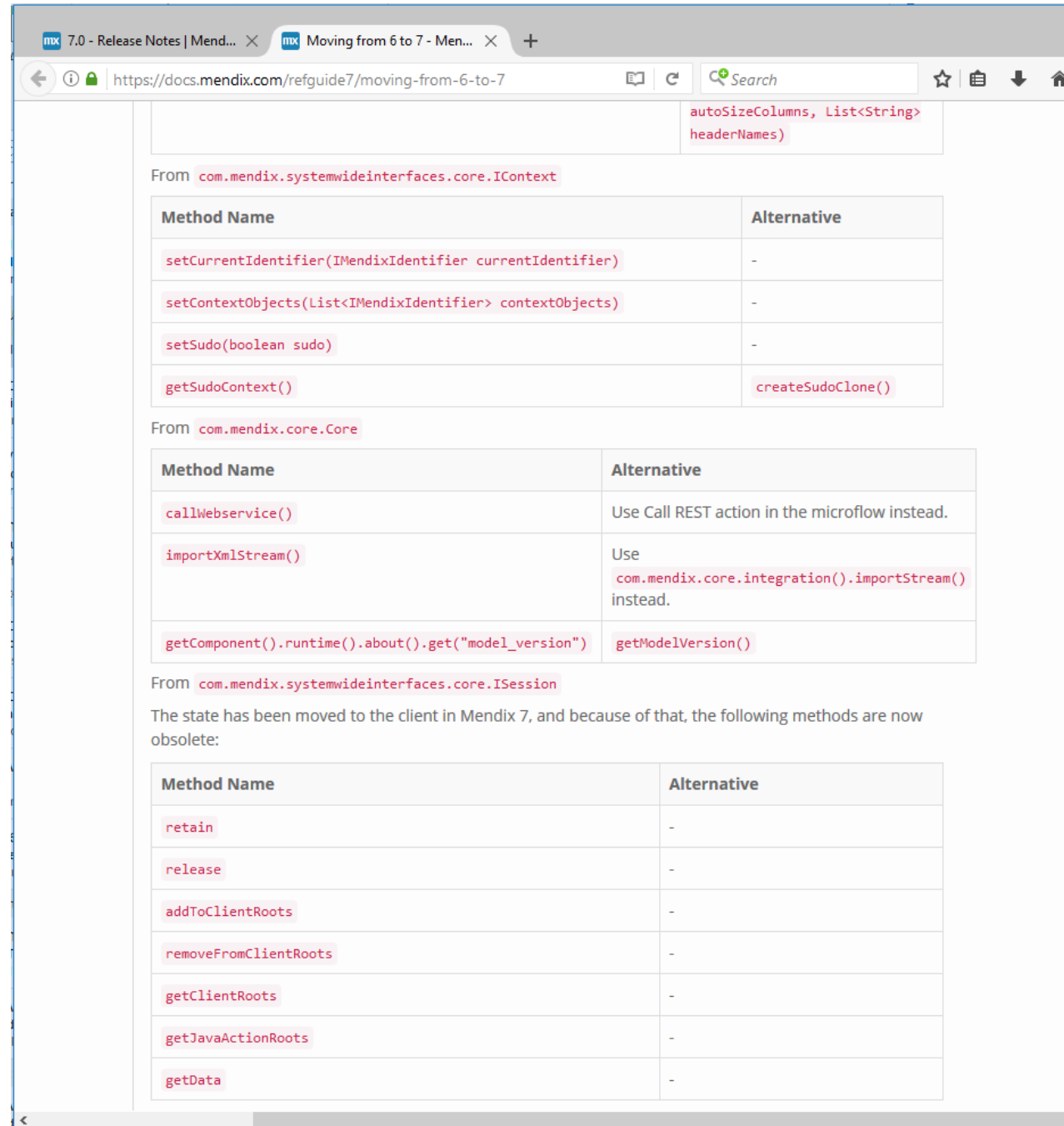
Migration

- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library



Migration

- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library



7.0 - Release Notes | Mendix ... Moving from 6 to 7 - Mendix ...

https://docs.mendix.com/refguide7/moving-from-6-to-7

autoSizeColumns, List<String> headerNames)

From `com.mendix.systemwideinterfaces.core.IContext`

Method Name	Alternative
<code>setCurrentIdentifier(IMendixIdentifier currentIdentifier)</code>	-
<code>setContextObjects(List<IMendixIdentifier> contextObjects)</code>	-
<code>setSudo(boolean sudo)</code>	-
<code>getSudoContext()</code>	<code>createSudoClone()</code>

From `com.mendix.core.Core`

Method Name	Alternative
<code>callWebservice()</code>	Use Call REST action in the microflow instead.
<code>importXmlStream()</code>	Use <code>com.mendix.core.integration().importStream()</code> instead.
<code>getComponent().runtime().about().get("model_version")</code>	<code>getModelVersion()</code>







From `com.mendix.systemwideinterfaces.core.ISession`

The state has been moved to the client in Mendix 7, and because of that, the following methods are now obsolete:

Method Name	Alternative
<code>retain</code>	-
<code>release</code>	-
<code>addToClientRoots</code>	-
<code>removeFromClientRoots</code>	-
<code>getClientRoots</code>	-
<code>getJavaActionRoots</code>	-
<code>getData</code>	-

Migration


















- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library

-  biz.aQute.bnd.bndlib.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles
-  com.mendix.json.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles
-  com.mendix.logging-api.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles
-  com.mendix.m2ee-api.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles
-  com.mendix.public-api.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles
-  javax.servlet.jar - C:\Program Files\Mendix\7.0.0\runtime\bundles

Migration

- ▶ Upgrade to the latest 6, i.e. 6.10.5
- ▶ Fix errors, warnings and deprecations
- ▶ Replace legacy layouts
- ▶ Upgrade App Store modules
 - Location of App Store modules in your project has moved to Project node
- ▶ Update your use of java APIs
 - Mendix API is more strict -> API may have been renamed or removed
 - Classloader is more strict -> You cannot use all the jars shipped with Mendix runtime, just the jars in userlib
 - Classloader only loads newest jar of a particular library

temp > Mx7Demo-main > userlib

Name	Date modified
 axiom-api-1.2.12.jar	3/20/2017 2:23 PM
 axiom-api-1.2.12.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 commons-lang-2.5.jar	3/20/2017 2:23 PM
 commons-lang-2.5.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 commons-logging-1.1.jar	3/20/2017 2:23 PM
 commons-logging-1.1.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 guice-2.0.jar	3/20/2017 2:23 PM
 guice-2.0.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 openid4java-0.9.8.jar	3/20/2017 2:23 PM
 openid4java-1.0.0.jar	3/20/2017 2:23 PM
 openid4java-1.0.0.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 org.apache.commons.lang3.jar	3/29/2017 3:48 PM
 org.apache.commons.lang3.jar.ObjectHandling.RequiredLib	3/29/2017 3:48 PM
 org.apache.httpcomponents.httpClient-osgi.jar	3/20/2017 2:23 PM
 org.apache.httpcomponents.httpClient-osgi.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM
 org.apache.httpcomponents.httpcore-osgi.jar	3/20/2017 2:23 PM
 org.apache.httpcomponents.httpcore-osgi.jar.AppCloudServices.RequiredLib	3/20/2017 2:23 PM

Migration – Impact of Stateless Runtime

- ▶ Every session is a persistent session
 - After logout it may take up to 30 seconds before the logout is visible on all runtime instances
 - You can configure this using `SessionValidationTimeout`
- ▶ NPE attributes need to have read access
- ▶ Autocommitted objects not supported in system sessions
- ▶ Sign-in microflow has been removed

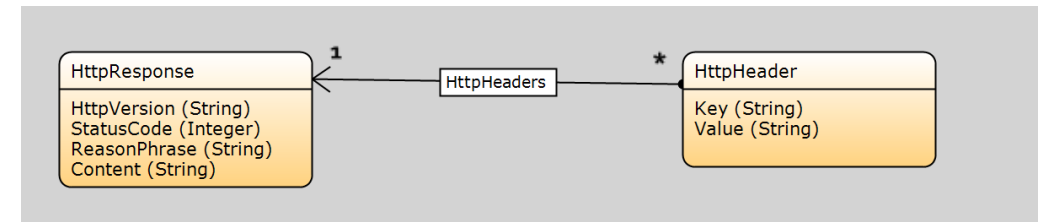
Rest

New Rest Features Mendix 7

- ▶ HTTP response metadata
- ▶ Optionally raw response payload
- ▶ **Access to headers**
- ▶ **Access to cookies (via headers)**
- ▶ **Access to status**
- ▶ **urlEncode & urlDecode**

The screenshot shows the 'Call REST' dialog box with the 'Response' tab selected. The configuration is as follows:

- Response handling:** Apply import mapping
- Import mapping:**
 - Mapping: BoxDemo.ListFoldersImportMapping
 - Parameter: (empty)
 - Range: All First Custom
- Output:**
 - Type: BoxDemo.Folders
 - Variable: Folders
- Error handling:**
 - Store message body in \$latestHttpResponse variable.



New Rest Features Mendix 7

- ▶ HTTP response metadata
- ▶ Optionally raw response payload
- ▶ Access to headers
- ▶ Access to cookies (via headers)
- ▶ Access to status
- ▶ **urlencode & urlDecode**

The image displays two screenshots from Mendix Studio. The top screenshot shows the 'Call REST' dialog box with the following configuration:

- Location: `https://maps.googleapis.com/maps/api/geocode/json?address={2}&key={1}`
- HTTP method: GET
- Timeout: 30 seconds
- Use timeout on request: No

The bottom screenshot shows a process flow diagram. It starts with an 'Address' widget, followed by a 'Call REST (GET)' widget. The 'Call REST' widget has two output parameters: 'ValidationResponse' and 'AddressValidation'. The flow then goes to a 'Log message (info)' widget, and finally to a 'Change 'Address' (ValidatedAddress, Longitude, Latitude)' widget. Both the 'Call REST' and 'Change Address' widgets have a 'Usage' icon above them.

The bottom screenshot also shows the 'Location' dialog box for the 'Call REST' widget, with the following configuration:

- Template: `https://maps.googleapis.com/maps/api/geocode/json?address={2}&key={1}`
- Parameters:

Index	Expression
{1}	@Main.GOOGLE_API_KEY
{2}	urlencode(\$Address/AddressSummary)

More Info

More info

- ▶ <https://docs.mendix.com/releasesnotes/desktop-modeler/7.0>
- ▶ <https://docs.mendix.com/refguide7/moving-from-6-to-7>
- ▶ <https://docs.mendix.com/refguide7/clustered-mendix-runtime>
- ▶ <https://docs.mendix.com/howtogeneneral/bestpractices/best-practices-for-app-performance-in-mendix-7>

Thank You!
QUESTIONS?