

Webinar - Kick start your widget development



Thank you for joining the Expert webinar "Kickstart your widget development". My name is Jelte Lagendijk, I am a JavaScript developer for the Community team at Mendix. At the Community team we want to enable our Community to create content. In this particular case, we want to enable you as a JavaScript developer to develop your own widgets.

For this webinar we expect you to know how a widget works, have technical knowledge about JavaScript and have a basic understanding what Node.js is.

Agenda

Agenda

- ▶ Scaffolding a widget in a minute
- ▶ Speeding up development using Grunt
- ▶ Updating existing widgets
- ▶ Deprecations (Bonus)
- ▶ Demo time!



This webinar will demonstrate new tooling: the use of the Yeoman widget generator and Grunt.

We'll not go in-depth on how a widget works. We have a tutorial on how to create your first widget:

<https://world.mendix.com/display/howto6/Creating+a+Basic+Hello+World+Custom+Widget>

Scaffolding (1)

Scaffolding (1)

- ▶ Previously Brackets + Mendix widget plugin
 - Visual and easy to use
 - Has a few bugs that are hard to fix
 - Less support – smaller community
- ▶ Introducing a new tool: Yeoman generator
 - Editor independent
 - Faster
 - Better maintainable



The Developing a widget using Brackets how-to can be found here:

<https://world.mendix.com/display/howto6/Getting+started+with+the+Widget+Development+Plugin+for+Adobe+Brackets>

Scaffolding 2

Scaffolding (2)

- ▶ Yeoman is a scaffolding tool running on top of Node.js



- ▶ Client uses generators as templates to scaffold various web applications or frameworks
 - <http://yeoman.io>
- ▶ Widget generator



- The documentation for the Yeoman widget generator is not available yet, this will be published soon. A quick how-to is provided on GitHub:

<https://github.com/mendix/generator-mendix/blob/master/README.md>

- You can find Yeoman at <http://yeoman.io>

Speed up development using Grunt (1)

Speed up development using Grunt (1)

- ▶ Grunt is a task runner, running on top of Node.js



- ▶ Define your own tasks in Gruntfile.js
 - <http://gruntjs.com>
- ▶ Out of the box tasks to speed up your widget development
 - We include the Gruntfile.js in our widget generator as well as many widgets on Github



Grunt is a task runner that can be found on <http://gruntjs.com>

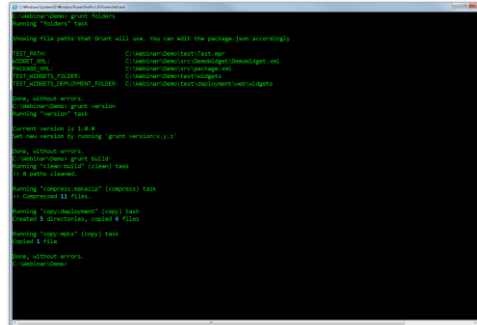
- Built on top of Node.js
- Has the ability to run predefined/custom defined tasks
- The Grunt client uses a single file, a Gruntfile.js, to run these tasks. Dependencies are installed from package.json
- The main advantage is that it doesn't depend on the editor you use. You can run this from the command-line, but, for a lot of editors there are plugins that can use Grunt as well.

Speed up development using Grunt (2)

Speed up development using Grunt (2)

▶ Available tasks

- build
- version
- start-modeler
- folders
- default (watch)

A screenshot of a terminal window with a black background and white text. The text shows the output of a Grunt command. It starts with a warning about file paths, followed by a list of tasks being executed: 'clean', 'copy:src', 'copy:dist', 'copy:images', 'copy:css', 'copy:scripts', 'copy:languages', 'copy:languages', 'copy:languages', 'copy:languages', 'copy:languages', 'copy:languages'. It then shows 'grunt version' outputting '1.4.9', followed by 'grunt build' outputting 'Building "compressed" (compress) task' and 'Building "start-modeler" (start) task'. The terminal ends with 'Done, without errors.'

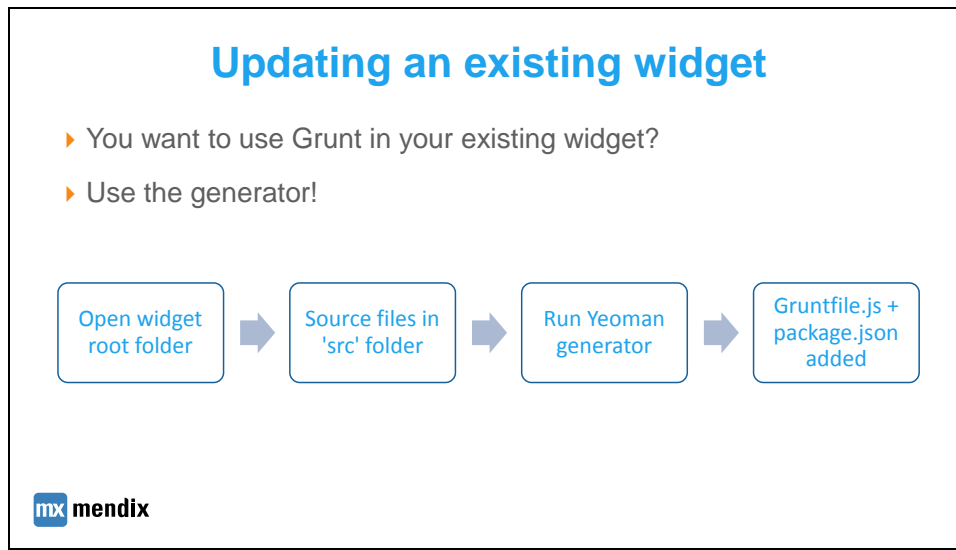
Running a tasks:

> **grunt <taskname>**

Available tasks:

- build – create an .mpk file that is used in the modeler and can be distributed
- version – check/change the version number of the widget
- start-modeler – start your modeler from Grunt
- folders – check which folders are used by Grunt
- default (watch) – watches for changes in your src folder and executes build & copy (used the most)

Updating an existing widget



The setup for these folders is the same for every widget:

- WidgetName <root folder>
 - src
 - package.xml
 - WidgetName <widgetfiles>
 - test <test-project>

See the AppStoreWidgetBoilerplate for a setup:

<https://github.com/mendix/AppStoreWidgetBoilerplate>

Deprecations (bonus)

Deprecations (bonus)

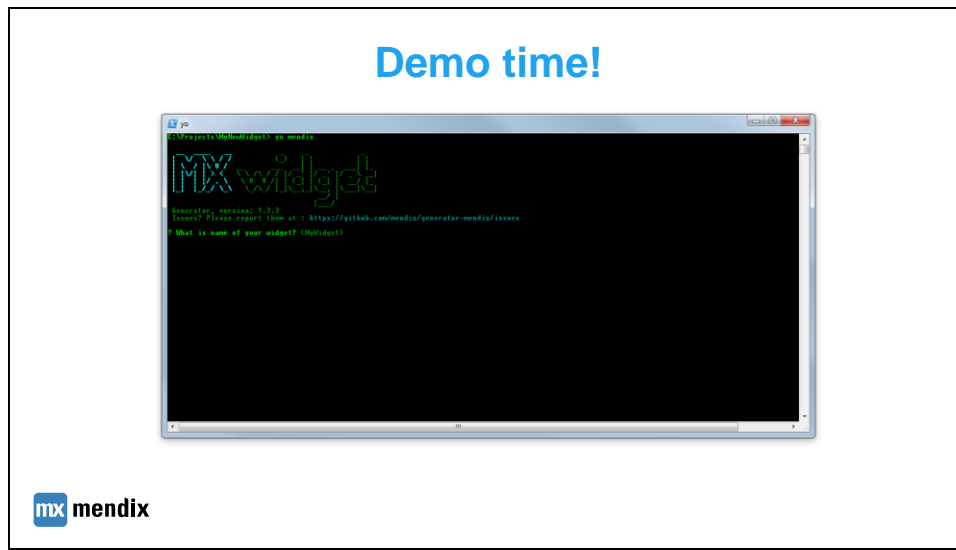
- ▶ We needed a tool to check for deprecations that would break widgets in MX6
- ▶ Built on top of Node.js
- ▶ Use this tool to test your current widgets:

mx-check-deprecations



The deprecation checker can be found at: <https://github.com/mendix/mx-check-deprecations>
This standalone client will let you quickly check if there are any deprecations in your widgets, for example in your widgets folder of your project.

Demo time!



Installing the generator

- Install Node.js from <http://nodejs.org>
- Open a terminal (Powershell or CMD) and check if Node is installed by typing: **node -v**
- Install the necessary tooling (Yeoman, Widget generator and Grunt client) by typing: **npm install -g yo generator-mendix grunt-cli**

Scaffolding a widget

- Create a new folder "MyWidget" and open this in your terminal
- Run the generator by typing: **yo mendix**
- Enter the necessary parameters, let the generator scaffold a new widget and automatically install the dependencies for Grunt

Updating a widget with the latest Gruntfile

- Make sure you have the default folder structure of a widget (source files should be in 'src')
- Run **yo mendix** in the root folder of the widget
- It will ask you if you are updating a widget, press **y** & Enter to confirm

Installing Grunt dependencies for a downloaded or cloned widget

- Make sure your root folder contains a **Gruntfile.js** and **package.json**
- Install the dependencies by typing **npm install**
- Now you can use Grunt in developing this widget

Using Grunt

- In the root folder of your widget, run a Grunt task by typing in your terminal: **grunt <taskname>**
- Let's build the .mpk file by running **grunt build**

- I want to update my version number: ***grunt version*** to check. Let's update it to version 1.1.0 by running ***grunt version:1.1.0***
- Let's open the test-project by running ***grunt start-modeler***
- Locally deploy your test-project and check if your widget works in the browser (We will use Chrome). Let's open the Chrome Inspector. (Right-click -> "Inspect")
- Run ***grunt*** (without a task name). Let Grunt watch for changes and automatically copy them to the deployment folder (and let this run in the background)
- Change some code of your widget, refresh the page in your browser and check the changes.

Tips using Grunt

- If you change any CSS or XML file in your widget, you need to synchronize your project directory (By pressing F4 in your modeler) and redeploy locally.

Deprecations

- Install the deprecation checker by typing in your terminal ***npm install -g mx-check-deprecations***
- Run this tool in your widgets folder by typing ***mx-check-deprecations***
- This will tell you which widgets it will check and where there are possible deprecations
- More information: ***mx-check-deprecations --help***