

# Being more efficient with VS Code

David Feldhoff – GWS

Tobias Fenster – 4PS

# Speaker intro

**Tobias Fenster**



Managing Partner at 4PS Germany



tobiasfenster on Twitter and LinkedIn

Blog [tobiasfenster.io](https://tobiasfenster.io)

**David Feldhoff**



Developer at GWS mbH in Germany



VS Code Ext. (AL CodeActions)

FeldhoffDavid on Twitter and LinkedIn



# Agenda

 Snippets


 Configurations

 Regular expressions

 Recommended extensions

 Combined example

 Build your own extension

 Q&A





# Snippets

# Snippets

- Snippets = Code templates with placeholders
- Bunch of snippets already exist, e.g. for
  - whole objects
  - Object parts
    - page actions / fields / parts, report data items / columns, ..
  - constructs
    - if, if/else, case, case/else, for, foreach, repeat, while
  - ...
- Command showing all snippets: “Insert Snippet”



# Custom Snippets

- <https://code.visualstudio.com/docs/editor/userdefinedsnippets>
- Start with extension “Snippet Creator” and then -> Configure User Snippets
- Tabstops:
  - **\$1, \$2** are tabstops. **\$0** is the last tabstop and ends the snippet mode
  - Can have placeholders
    - Hard-coded values like **\${2:true}**
    - With variables like **\${1:\$CLIPBOARD}**
    - With choices like **\${1|Codeunit,Page,Table,Report|}**
  - Supports replace-transformations with regular expressions (later more)



# Excursion: Multi-Cursor Shortcuts

- Shown:
  - Add Selection To Next Find Match      Ctrl + D
  - Select All Occurrences of Find Match      Ctrl + Shift + L
- Other possibilities:
  - Add Cursors to Line Ends      Shift + Alt + I
  - Add Cursor Above      Ctrl + Alt + UpArrow
  - Add Cursor Below      Ctrl + Alt + DownArrow
  - Place multiple Cursors manually      Alt + Mouse click





# Configurations



# Move and code faster

- Fast scrolling: Alt
- Go to symbol in file: Ctrl + Shift + O
- Breadcrumbs: Open and select: Ctrl + Shift + .
- Navigate to line: Ctrl + G
- Configurable:
  - Default list
  - Add / change shortcuts
- Configurations
  - Can sync across machines
  - Have profiles for different setups



# Work faster

- Interesting configurations that make you more efficient:
  - Merge editor for conflicts (`git.mergeEditor`)
  - Format on save mode (`editor.formatOnSaveMode`)
  - Simple files dialog (`files.simpleDialog.enable`)

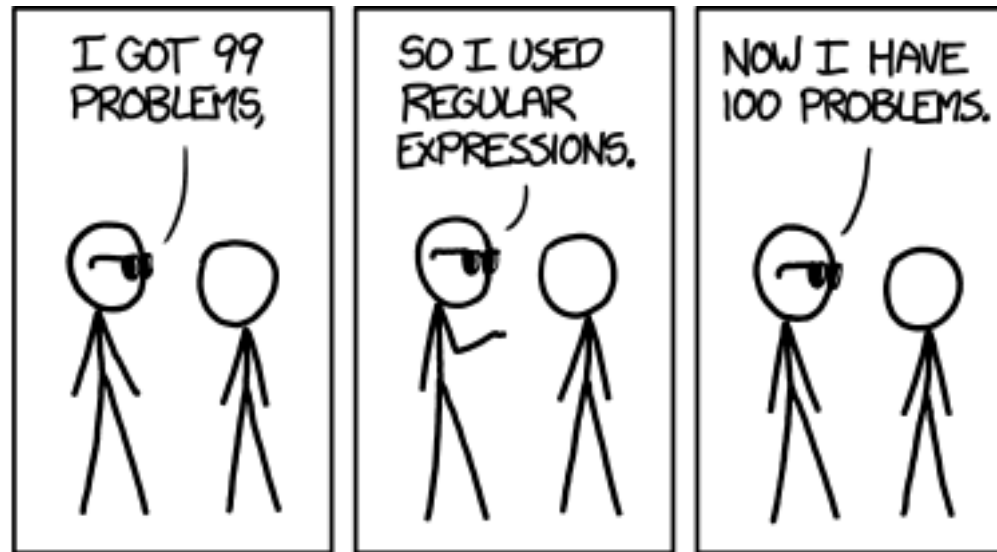


The background of the slide is a photograph of a modern, multi-story building with a glass facade. The image is divided into three vertical color bands: red on the left, orange in the middle, and green on the right. The text 'Regular expressions (Regex)' is centered across the middle band in a white, bold, sans-serif font.

# Regular expressions (Regex)

# Regex

- Regular expression ("regex")
  - Search for a specific pattern in a text
  - More powerful replace functionality



```
/^[a-zA-Z_\-]+\@((([a-zA-Z_\-]+\.)+[a-zA-Z]{2,4})$)/
```



# Regex

- Why so scared? What we have to learn is only 7 or 8 things:
  - Special characters
    - Groups
    - Character classes
      - Special character classes
    - Quantifiers
- Demo: Get all parts of a table definition (object type, id and name)
  - E.g. table 18 Customer



# Show-/Usecases of regex

- Create permission set to show regex-replace
- Find relevant test library procedures for creating sales lines



The background image shows a modern, multi-story building with a glass facade and a prominent tower-like structure on the left. The image is divided into three vertical color zones: red on the left, orange in the middle, and green on the right. The text "Recommended extensions" is centered across the middle of the image in a large, white, bold font.

# Recommended extensions

# Before I start – THANKS to all the creators!



**AZ AL Dev Tools/AL Code Outline** v3.0.35

Andrzej Zwierzchowski

AZ AL Development Tools: AL code outline, object browser, object creators

{AL}

Object  
Designer

**AL Object Designer** v0.2.4

Márton Sági

Object Designer optimized for (Dynamics 365 Business Ce...



**XLIFF Sync** v1.3.0

Rob van Bekkum

A tool to keep XLIFF translation files in sync.



**AL Variable Helper** v2.4.0

Rasmus Aaen

Intellisense to assign cor



**AL Studio** v0.9.934 Preview

Dynasit Solutions Kft.



**AL Extension Pack** v1.0.19

waldo

Everything you need for AL development (Microsoft Dyna...

## Azure DevOps Simplify

tfenster

An Extension to make working with Azure DevOps in VS Code more efficient



**AL Toolbox** v2.0.22

Bart Permentier

AL Language Regions and Snippets



**AL Object ID Ninja** v2.9.1

Vjeko.com

Manage object IDs in multi-user environments with mind-boggling simplicity.



BC Linter Cop

**BusinessCentral.LinterCop** v0.1.4 Preview

Stefan Maron

Provides Linting for the AL Language (Business Central)



**NAB AL Tools** v1.26.0

NAB Solutions AB

Development and translation management tool for AL language and Microsoft Dynamics...



**AL Navigator** v0.5.0

Waldemar Brakowski

Create variables and parameters like a boss. Navigate through al files with aweso



**AL Test Runner** v0.7.13

James Pearson

Test runner for AL tests for Business Central development



**AL CodeActions** v1.0.20

David Feldhoff

Adds codeactions to the AL language



**AL Structure Creator** v0.3.0

EdySpider

Create folder structure for AL projects



**AL Object Helper** v2.2.10

DSaladin

Quick and easy access to any al object in your project



**AL XML Documentation** v1.1.7

365 business development


Generate XML documentation comments for AL source code and create markdown docu...





# , but useless Get direct support inside VS Code

- Power Mode extension



**Power Mode** v3.0.2  
Cody Hoover | 668,867 | ★★★★★ (71)  
Your code is powerful, unleash it! The extension made |  
[Disable](#) [Uninstall](#) [Refresh](#) [Settings](#)  
This extension is enabled globally.

[Details](#) [Feature Contributions](#) [Runtime Status](#)

## VSCODE POWER MODE!!!

Power Mode is one of the most requested extensions for VS Code.  
Unfortunately, they said it couldn't be done...

[GitHub, Inc. \[US\]](#) | <https://github.com/Microsoft/vscode/wiki/Requested-Extensions>

**Currently not implementable**

- [activate-power-mode](#) - not possible without DOM access in the API

However, after seeing [this list](#) and realizing that VS Code was the only modern editor without it, I knew I had to try. I couldn't let VS Code live in the shadow of



# Create/extend objects faster

- Object Wizards (AZ AL Dev Tools)
- Add multiple fields (AZ AL Dev Tools)
- Assign IDs in multi-user repos (AL Object ID Ninja)



# Cleanup your code faster

- Code transformation commands (AZ AL Dev Tools)
  - Remove with-usage
  - Sort variables
  - Remove unused variables
  - Add parentheses
  - ...
- Disable specific warnings with pragma (AL Toolbox)
- Get more code cops to comply with :)  
(BusinessCentral.LinterCop)



# Navigate and analyze faster

- Object explorer (AZ AL Dev Tools, AL Object Designer, AL Studio)
- Navigate to objects (AL Object Helper)
- Navigate to specific places in document (AL Navigator)
- Analyze faster
  - GitLens Views (File/Line History)
  - VS Codes Timeline-View (Local History)



# Code faster (1)

- Conversions
  - Extract procedures (AL CodeActions)
  - Convert text to labels (AL CodeActions)
  - Convert Options to enums (AL CodeActions)
  - Create interfaces based on codeunit (AZ AL DevTools)



# Code faster (2)

- Code faster: Create Variables (AL Variable Helper & AL Navigator)
- Code faster: Work with procedures (AL CodeActions)
  - Create procedures/publishers
  - Add parameters to procedures



# Demo: combine snippets, regex, VS Code extensions

- Complete test procedure-snippet



# Extended Backus-Naur form of snippets

any ::= tabstop | placeholder | choice | variable | text

tabstop

::= '\$' int

| '\${' int '}'

| '\${' int transform '}'

placeholder

::= '\${' int ':' any '}'

choice

::= '\${' int '|' text (',' text)\* '|' }

variable

::= '\$' var | '\${' var '}'

| '\${' var ':' any '}'

| '\${' var transform '}'

transform

::= '/' regex '/' (format | text)+ '/' options

format

::= '\$' int | '\${' int '}'

| '\${' int ':' '/upcase' | '/downcase' | '/capitalize' | '/camelcase' | '/pascalcase' '}'

| '\${' int ':' '+' if '}'

| '\${' int ':' '?' if ':' else '}'

| '\${' int ':' '-' else '}' | '\${' int ':' else '}'

regex

::= JavaScript Regular Expression value (ctor-string)

options

::= JavaScript Regular Expression option (ctor-options)

var

::= [\_a-zA-Z] [\_a-zA-Z0-9]\*

int

::= [0-9]+

text

::= .

if

::= text

else

::= text

**Result:**

`\${1 transform}`

`\${1 /regex/format/options}`

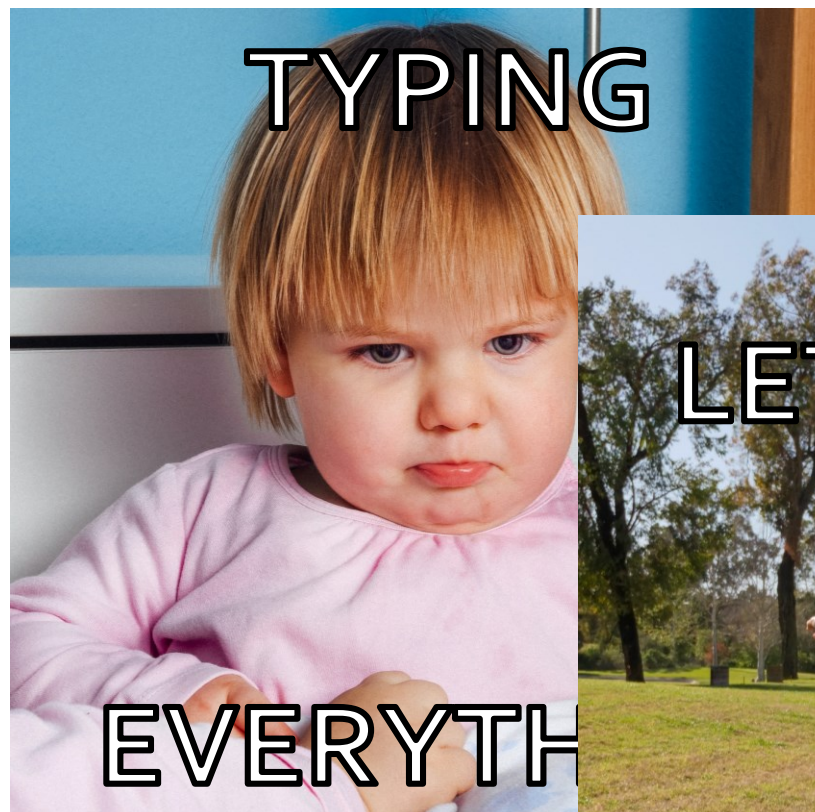
`\${1 /.\*format/}`

`\${1 /.\*/\${0:/pascalcase}/}`





# Code faster: GitHub Copilot





**Build your own extension**

# Build your own extension

- What is possible?
  - Change the look of VS Code with a color or file icon theme – Theming
  - Add custom commands – Extending the functionality
  - Add custom components & views in the UI – Extending the Workbench
  - Create a Webview to display a custom webpage built with HTML/CSS/JS – Webview Guide
  - Support a new programming language – Language Extensions Overview
  - Support debugging a specific runtime – Debugger Extension Guide
- Snippets in extensions
- Extension packs
- Extend extensions
- Get a pet to run around your editor...



# The elements of an extension

- **Activation** events: What makes your extension come alive – onCommand, onDebug, onView, onStartupFinished, ...
- **Contribution** points: What does your extension add – commands, debuggers, keybindings, viewContainers and views, ...
- **Code**: What does your extension do – call VS Code APIs, other extensions, your own logic, ...
- Important files:
  - **packages.json**: extension manifest incl. **contribution points and activation events**
  - **extension.ts**: main extension code



# World debut: Work faster with Azure DevOps Simplify

- Our little extension created for this session
- Helps with
  - Assigning commits to work items in two different ways
  - Creating branches linked to work items
- If you want even more: COSMO Alpaca as the (paid) efficiency helper for everything Azure DevOps and containers!
  - Find out more on the COSMO Marketplace or at the COSMO booth



# Work more efficient with your own extension

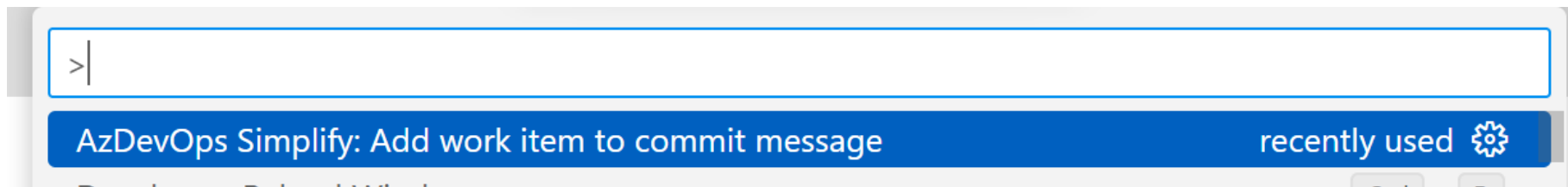
- Let's take a look at how we created the Azure DevOps Simplify extension
- yo code!
- GitHub Copilot
- Type code 😊



# Work more efficient with your own extension



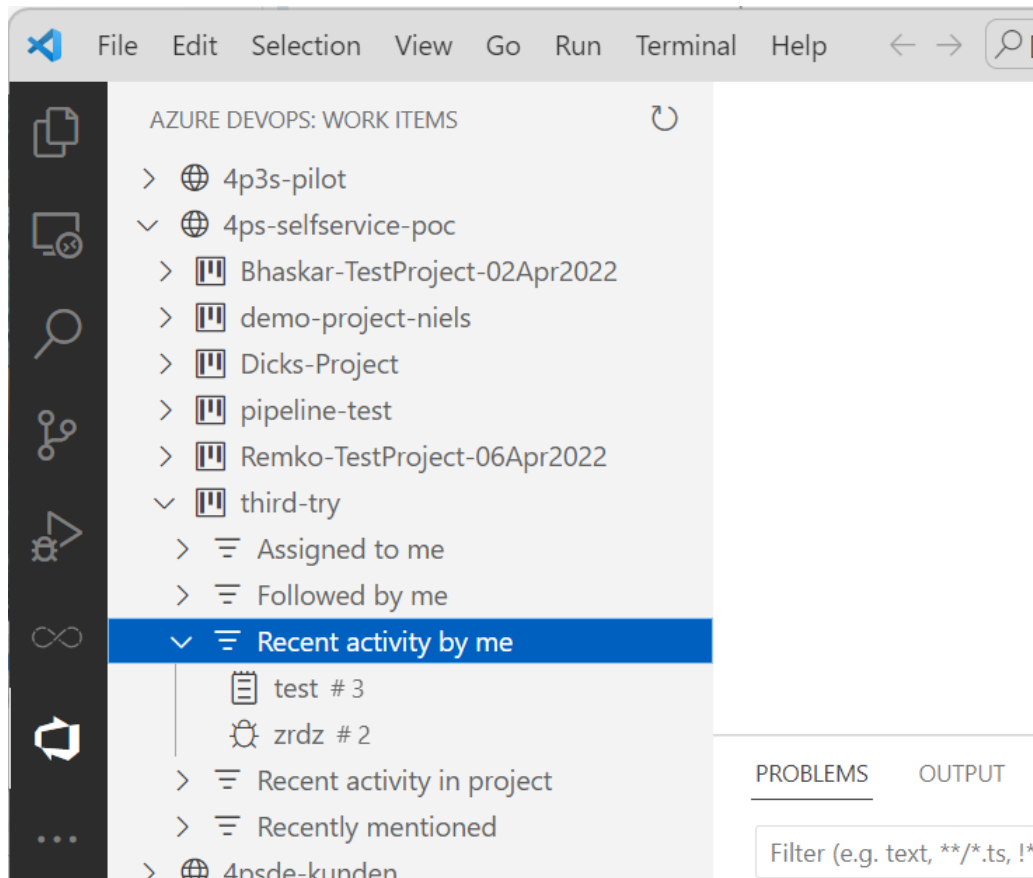
- Initial:
  - Use Yeoman to scaffold the basics of the extension
- First step, to make it faster:
  - Setup dependencies and add backend connecting code (not directly relevant for today)
- Second step
  - Contribute a command, register it as activation event and implement it





# Work more efficient with your own extension

- Third step:
  - Contribute a view container and view, register it as activation event and implement it








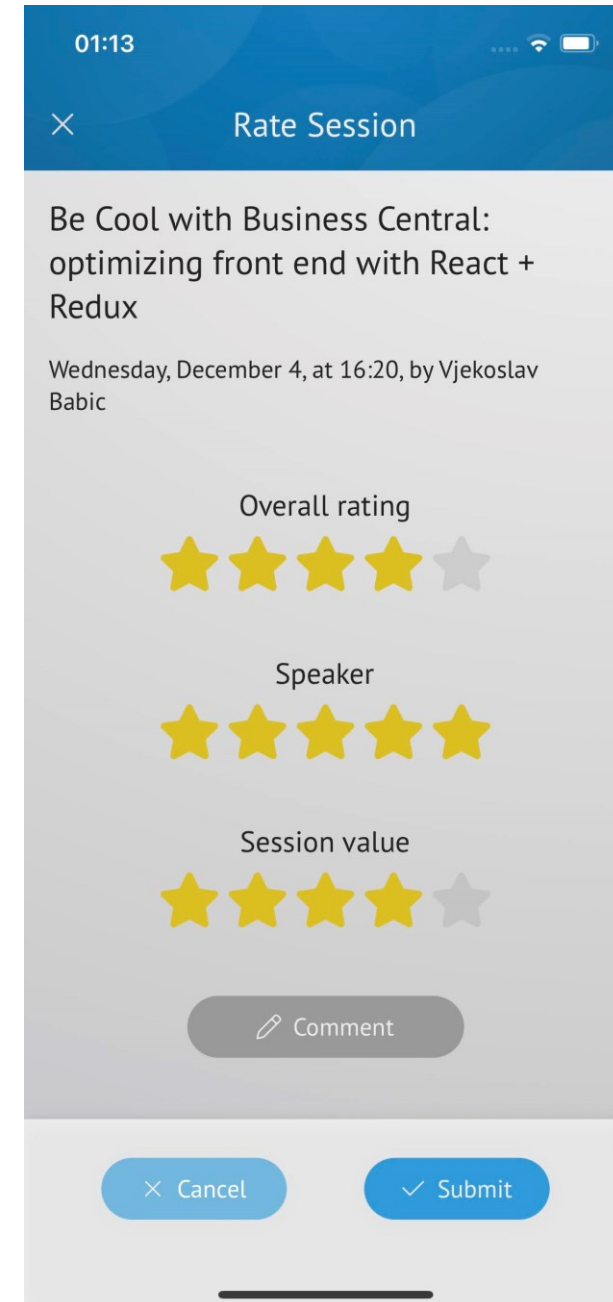
**Which questions can we  
try to answer?**

# Give us Feedback !

 Hope you enjoyed this session!

 Please help us improve and give session feedback in the App!

 **Thank you !**



The screenshot shows a mobile app interface for rating a session. At the top, the status bar shows the time 01:13 and battery level. The app header is blue with a close button (X) and the title 'Rate Session'. The main content area has a light gray background and displays the session title 'Be Cool with Business Central: optimizing front end with React + Redux', the date and time 'Wednesday, December 4, at 16:20', and the speaker 'by Vjekoslav Babic'. There are three rating sections: 'Overall rating' with 5 stars (4 yellow, 1 gray), 'Speaker' with 5 stars (all yellow), and 'Session value' with 5 stars (4 yellow, 1 gray). Below the ratings is a 'Comment' button with a pencil icon. At the bottom, there are 'Cancel' and 'Submit' buttons.

01:13

×

Rate Session

Be Cool with Business Central:  
optimizing front end with React +  
Redux

Wednesday, December 4, at 16:20, by Vjekoslav  
Babic

Overall rating

★★★★☆

Speaker

★★★★★

Session value

★★★★☆

✎ Comment

× Cancel

✓ Submit

