

# Development with Visual Studio

## Objectives:

- Install Visual Studio Code
- Describe Visual Studio Code Integrated Terminal
- Describe VSCode AEM Sync
- Install VSCode AEM Sync
- Test VSCode AEM Sync Extension



# Choose the right IDE

Integrated Development Environments (IDEs) are used to speed up development on projects. Ultimately, it's up to the developer and organization standards to choose.

Common Java IDEs used for AEM development

- Eclipse
- Visual Studio Code
- IntelliJ

Common Frontend IDEs for AEM development

- Visual Studio Code
- Brackets
- Basic text editors (Notepad++, bbedit, etc)

Depending on the project size and number of developers, it's not uncommon to use many different IDEs among different teams and initiatives.

# Key AEM Development features

Common features an IDE should have for AEM development

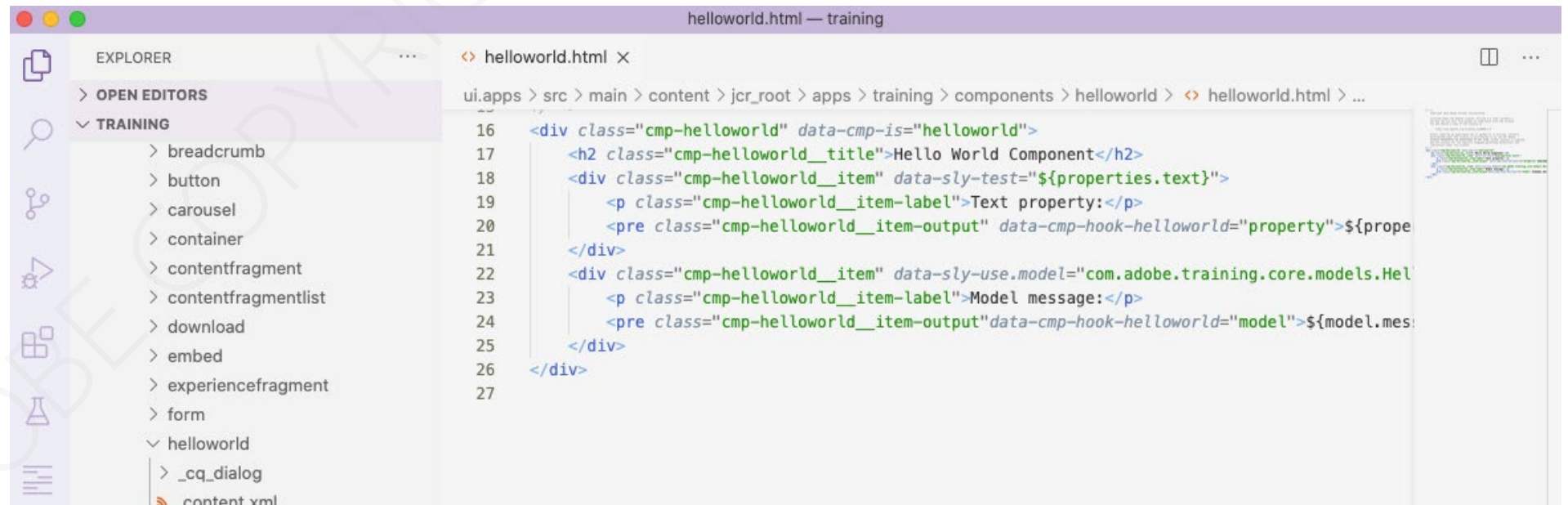
- Java Support
- Node/NPM support
- Maven
- Junit test runner
- Remote Java Debugging
- Dependency management and autocomplete
- AEM Project Sync

Nice to have

- JCR Properties view (.content.xml files)
- Git support

# Visual Studio Code

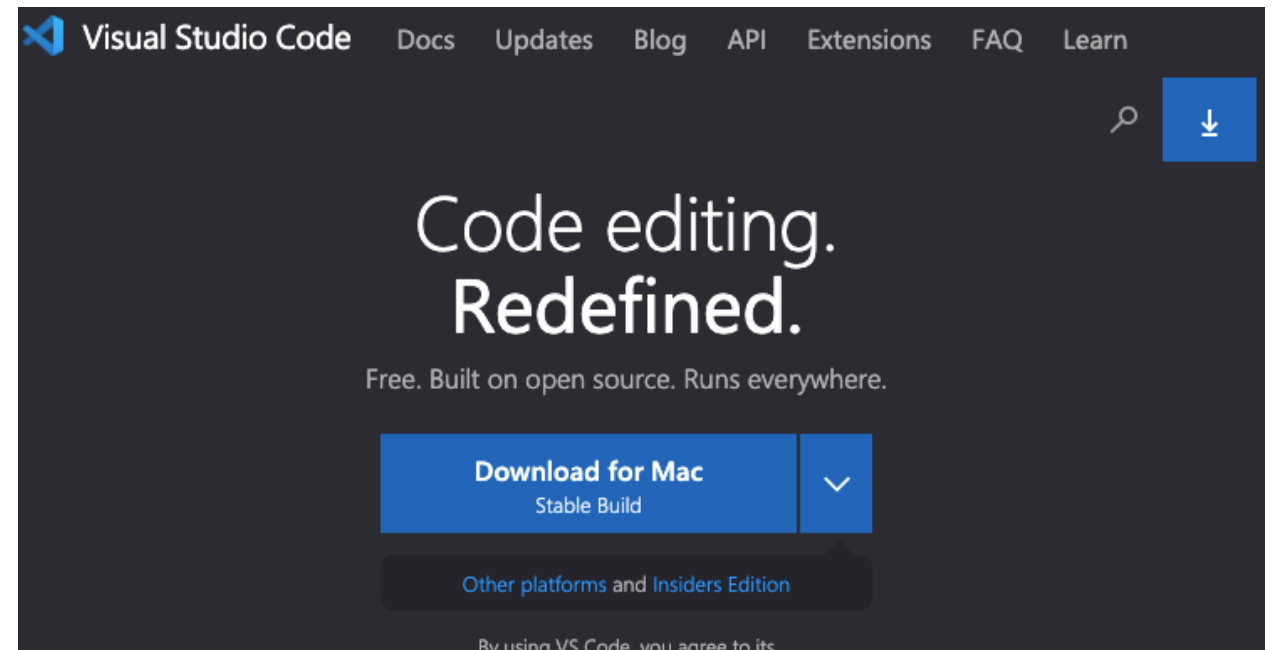
- Lightweight and a powerful source code editor
- Has built-in support for JavaScript, TypeScript, and Node.js



# Visual Studio Code Installation

- VS Code is a free editor
- Runs on the following operating systems:
  - macOS
  - Linux
  - Windows

<https://code.visualstudio.com>





## Exercise

# Exercise 1: Install Visual Studio Code (Local only)

ADLS Readytech environments already have Visual Studio Code installed

This exercise only needs completed if you are working locally

# Maven Use | Options

Deploying an entire project or module requires Maven. Two common options

- Maven command line outside Eclipse

```
[INFO] wetrain ..... SUCCESS [ 0.308 s]
[INFO] We.Train - Core ..... SUCCESS [ 4.234 s]
[INFO] We.Train - Repository Structure Package ..... SUCCESS [ 0.619 s]
[INFO] We.Train - UI apps ..... SUCCESS [ 3.957 s]
[INFO] We.Train - UI content ..... SUCCESS [ 1.594 s]
[INFO] We.Train - All ..... SUCCESS [ 6.018 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
```

- VSC Integrated Terminal

The screenshot shows the VSC Integrated Terminal interface. The 'TERMINAL' tab is selected and highlighted with a red box. The terminal output shows the Maven build process for 'wetrain' and its sub-modules, all of which completed successfully. The command 'mvn clean install -PautoInstallSinglePackage' is entered at the prompt.

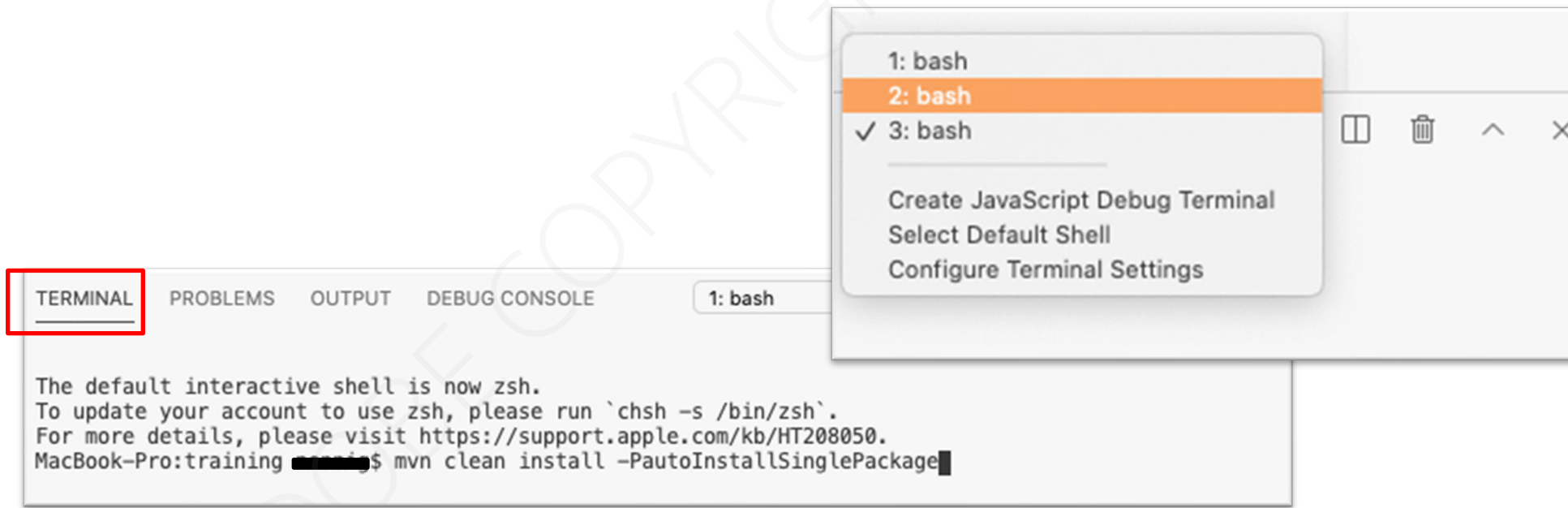
```
1: bash
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro:training ~$ mvn clean install -PautoInstallSinglePackage
```

*Either option can be used in training*

# Visual Studio Code Integrated Terminal

## Features of Integrated Terminal

- Can be opened initially starting at the root of your workspace
- Creates a separate window
- Multiple terminals option

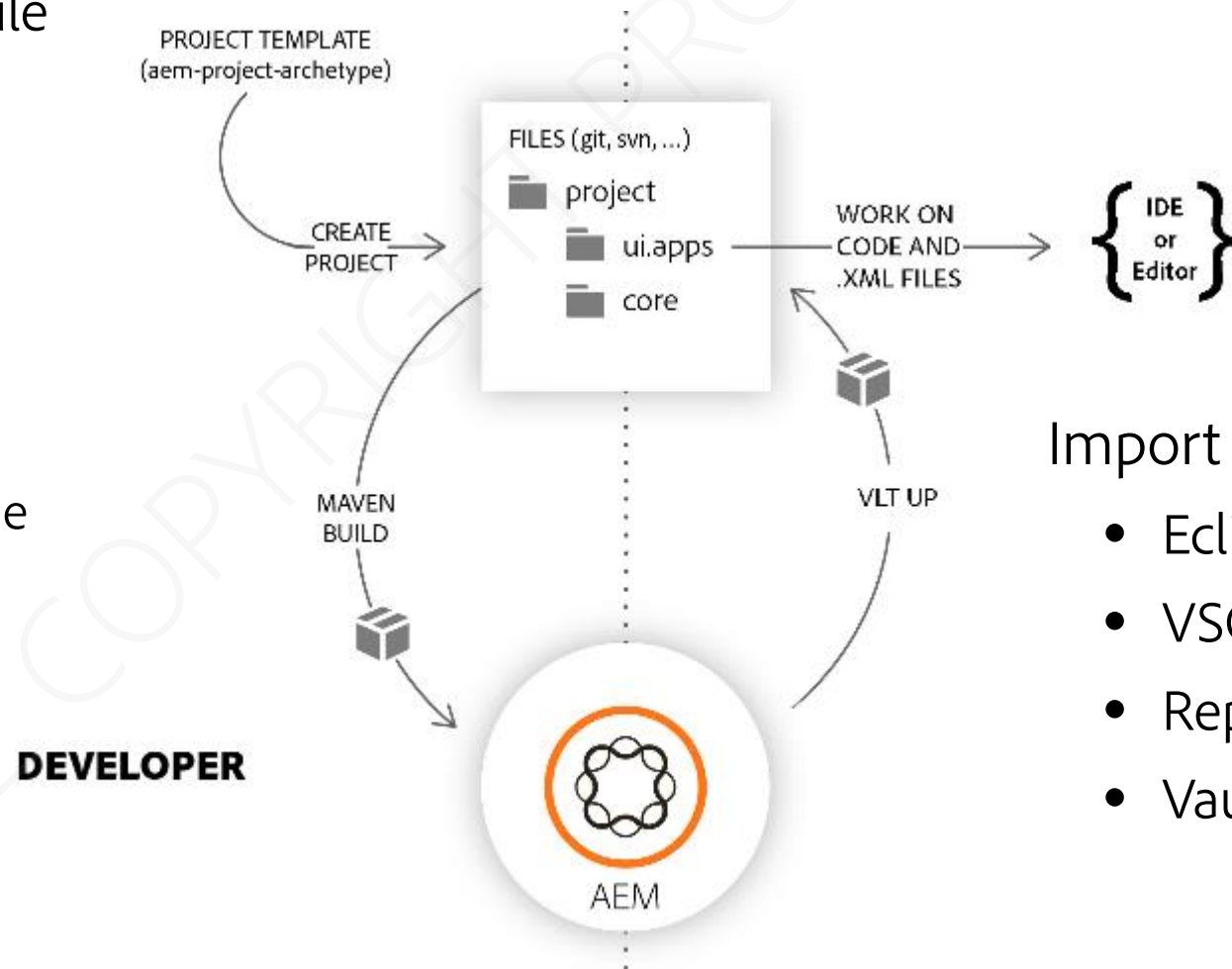




# AEM Project Sync | Common Options

## Export to AEM

- Entire project or module
  - Maven command
  - M2Eclipse extension
- Individual Files
  - Eclipse sync plugin
  - VSCode AEM sync
  - AEMFED command line
  - Repo command line



## Import from AEM

- Eclipse sync plugin
- VSCode AEM sync
- Repo command line
- Vault command line

# VSCode AEM Sync

## Code and content synchronization

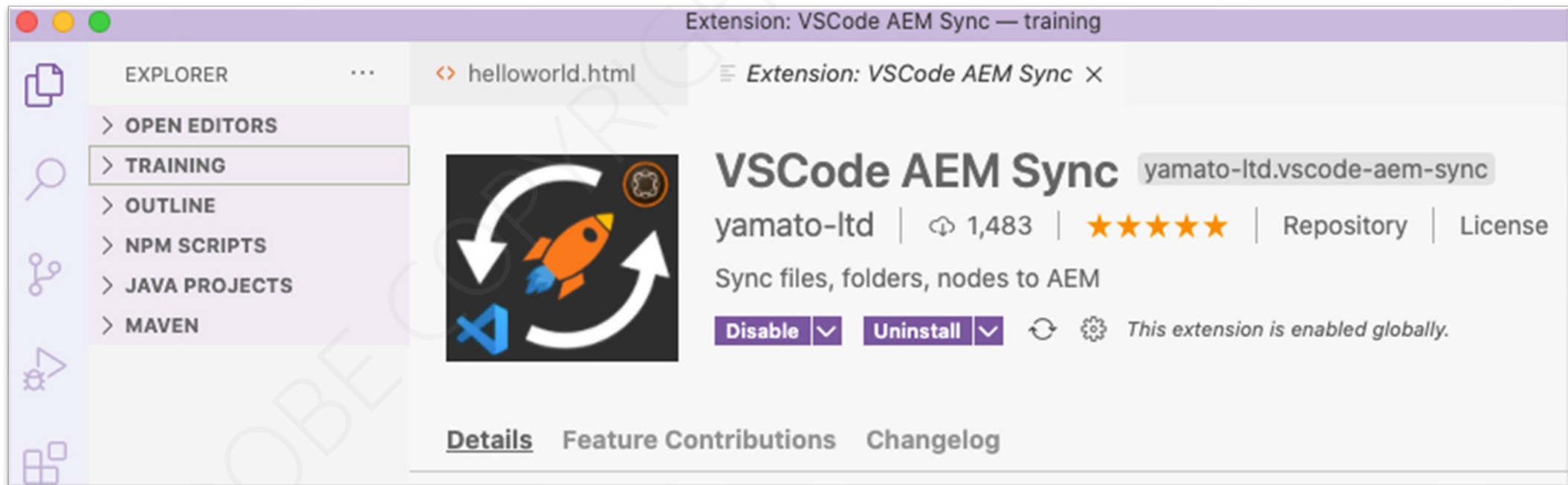
- Pushes from a local Maven project to a local AEM
- Only used for local development
- Sync is one way. Maven project --> AEM

## Benefits

- There is no vault dependency.
- Can push the content to multiple services at the same time
  - For example: author and publish
- Sync on file change(s)/save
- Full status output of sync

# VSCode AEM Sync | Installation

- Works on Windows, Linux, and macOS
- To install:
  - Go to <https://marketplace.visualstudio.com/items?itemName=yamato-ltd.vscode-aem-sync>





## Exercise

### Exercise 2: Install VSCode Sync (Local only)

ADLS Readytech environments already have Visual Studio Code installed

This exercise only needs completed if you are working locally

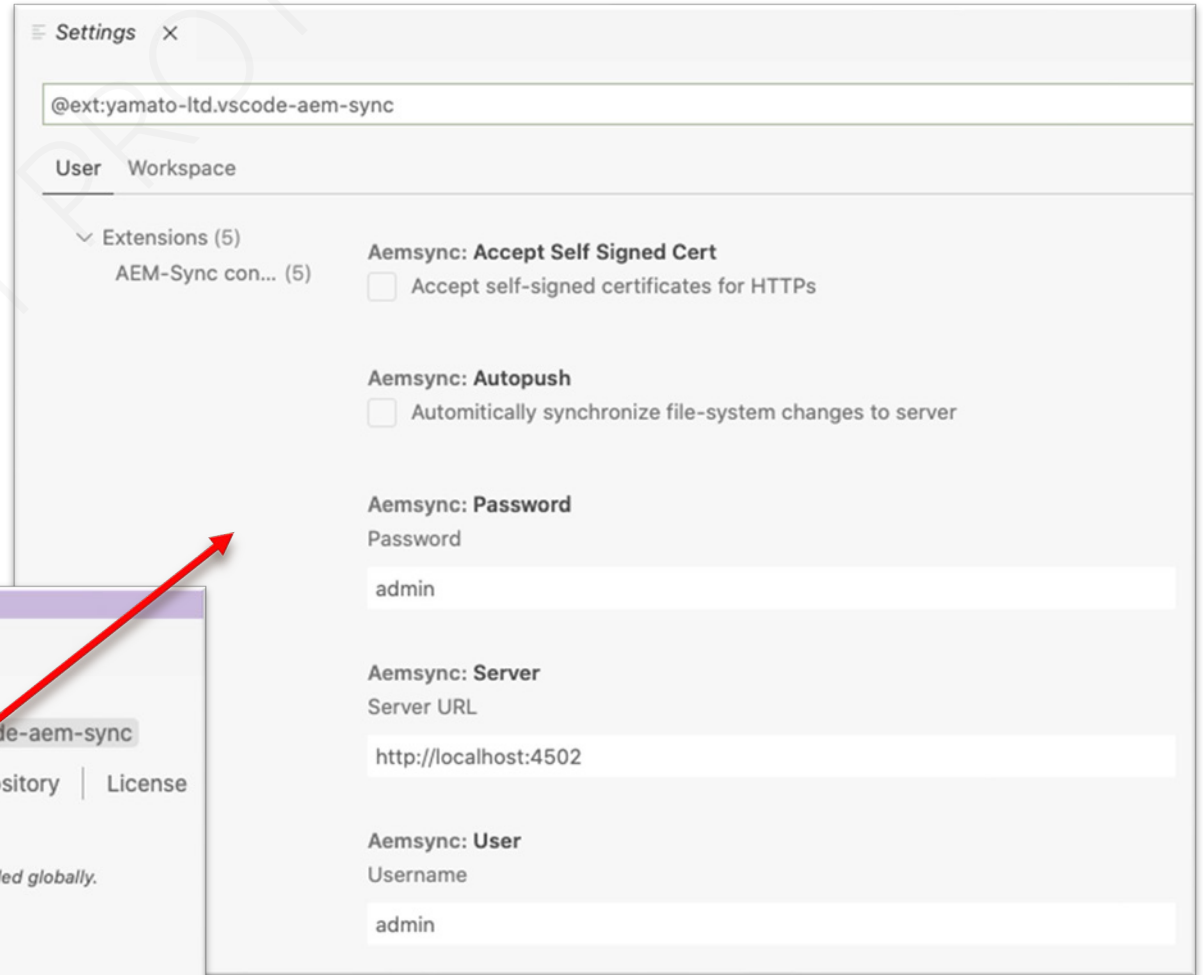
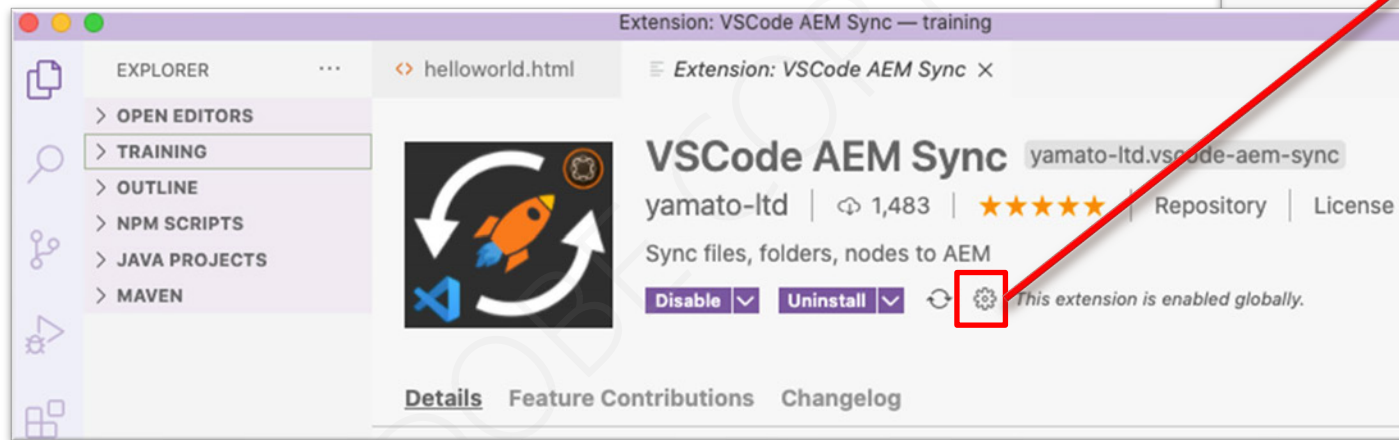
# VSC AEM Sync | Configuration

Accept self-signed certs

Auto sync of saved files to AEM

AEM Server info

- User, password, and host



# VSCode AEM Sync | Auto sync files on save

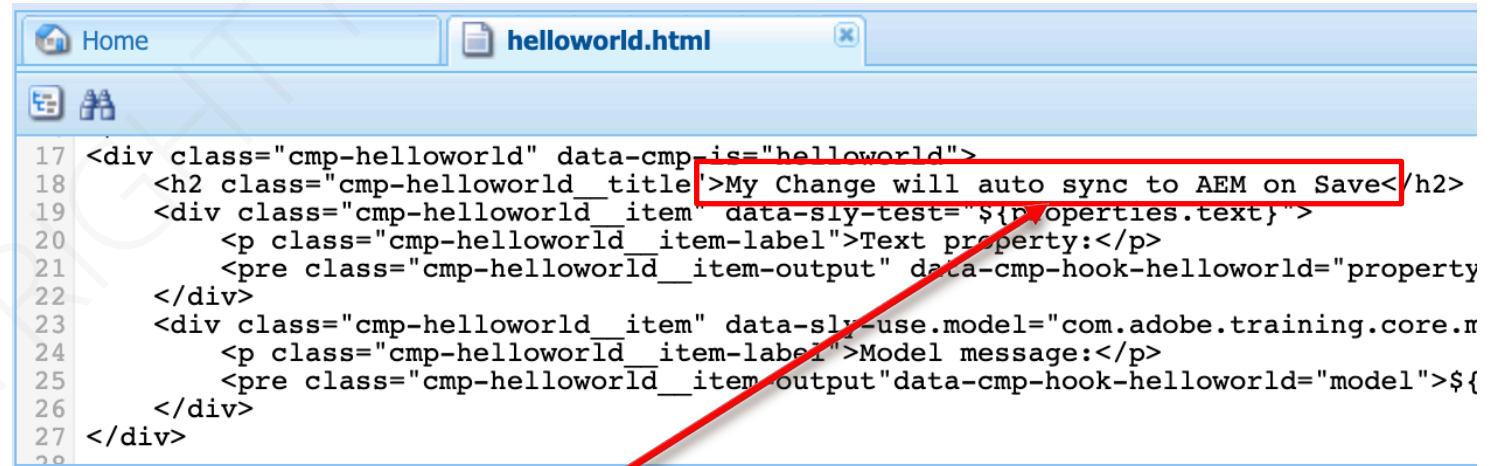
Make sure the extension is enabled

- Auto-push is on

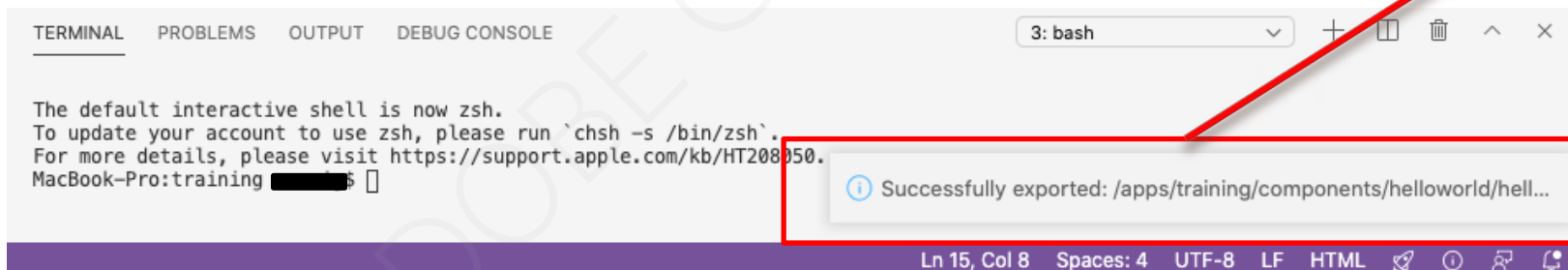
Make changes to the file(s)

- Save the file(s)

Verify the changes in AEM



```
17 <div class="cmp-helloworld" data-cmp-is="helloworld">
18   <h2 class="cmp-helloworld_title">My Change will auto sync to AEM on Save</h2>
19   <div class="cmp-helloworld_item" data-sly-test="{properties.text}">
20     <p class="cmp-helloworld_item-label">Text property:</p>
21     <pre class="cmp-helloworld_item-output" data-cmp-hook-helloworld="property"
22   </div>
23   <div class="cmp-helloworld_item" data-sly-use.model="com.adobe.training.core.m
24     <p class="cmp-helloworld_item-label">Model message:</p>
25     <pre class="cmp-helloworld_item-output" data-cmp-hook-helloworld="model">${
26   </div>
27 </div>
```



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
3: bash
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Pro:training $
```

Successfully exported: /apps/training/components/helloworld/hell...

Ln 15, Col 8 Spaces: 4 UTF-8 LF HTML



# VSCode AEM Sync | Import from AEM

Make sure the extension is enabled

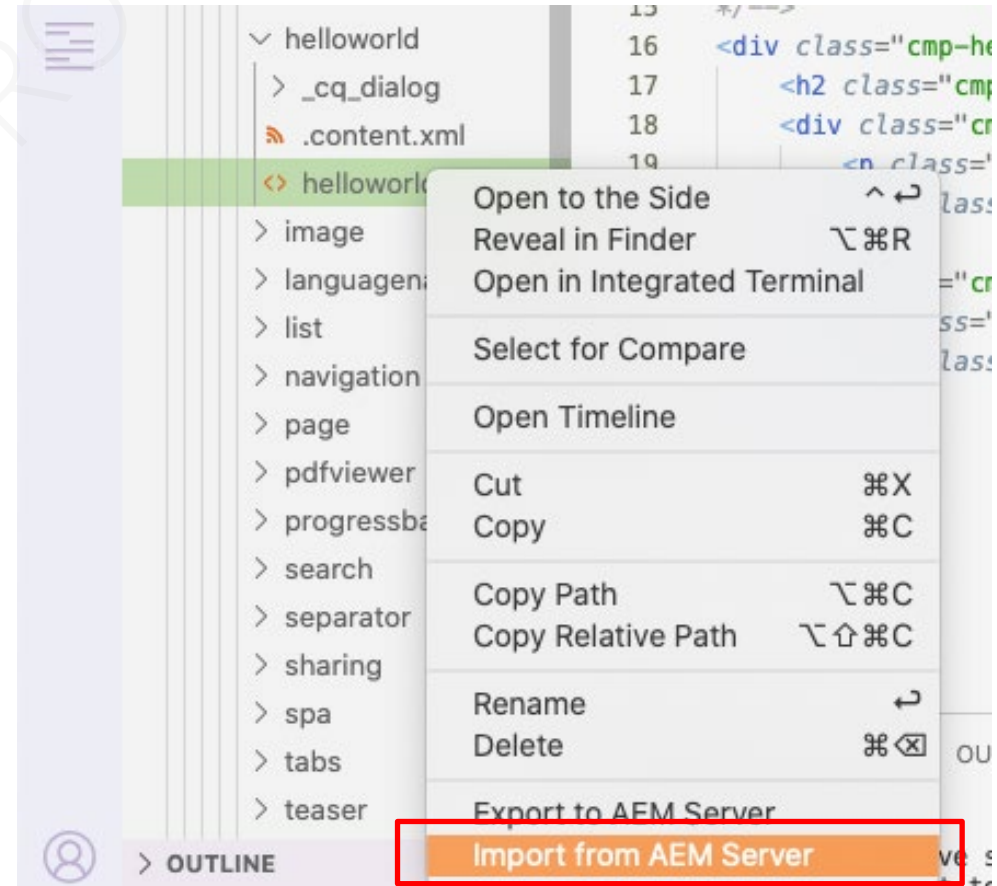
- Auto-push is on

Import CRXDE Lite Activities

- Dialogs structures reused
- Overlay nodes from /libs

Import skeleton content back to the project

- Pages, experience fragments, assets
- Configurations





Exercise

## Exercise 3: Import an AEM Project into VS Code

## Exercise 4: Synchronization tools for VSCode

Task 1 – Verify auto push is enabled

Task 2 - Sync AEM Content





## Key takeaways

### Key Takeaways:

- Visual Studio Code
  - Lightweight, powerful source code editor
- VSCode AEMSync
  - Code and content synchronization for AEM
  - Pushes the content to AEM services upon a file change
  - Used for local development