

# Day 17 - Composer in your module to load PHP libraries

## Objective

- In this session, we will see how we could leverage composer, and composer\_manager to load PHP libraries / SDKs in our custom modules

## Primary Tutorial

- <https://drupalcommerce.org/blog/31782/managing-d8-module-dependencies-new-composer-manager>
- <https://bojanz.wordpress.com/2015/09/18/d8-composer-definitive-intro/>

## Exercise

- On a fresh D8 instance, create a custom module
- Update the module's composer.json file to include this library - <https://packagist.org/packages/guhelski/forecast-php>
- Install [Composer Manager Module](#), run the init script. Run `composer drupal-update` such that the mentioned library is fetched to the vendor folder and autoloaded and hence is available for use in your module (This might not be required post 8.1.x???)
- Post 8.1.x, you could directly run `composer update` from docroot (?)
- Build a custom block with a [configuration form that takes latitude and longitude](#) in the configuration form
- The block, when enabled should show the forecast for the configured location by in simple text as "Forecast is XXXXX with temperature of XXX dec C".



- This forecast information is retrieved using [Forecast wrapper library that we included](#).
- API Key you could use = **7411b0e6d5e0c99fbd7405fd6de00cd5** (Alternatively, you could register on [forecast.io](#) for the key)

## Bonus Exercise

- Explore why you would need composer\_manager module in the first place since you could just simply run `composer install` from the module folder
- Explore what the init script of composer\_manager does