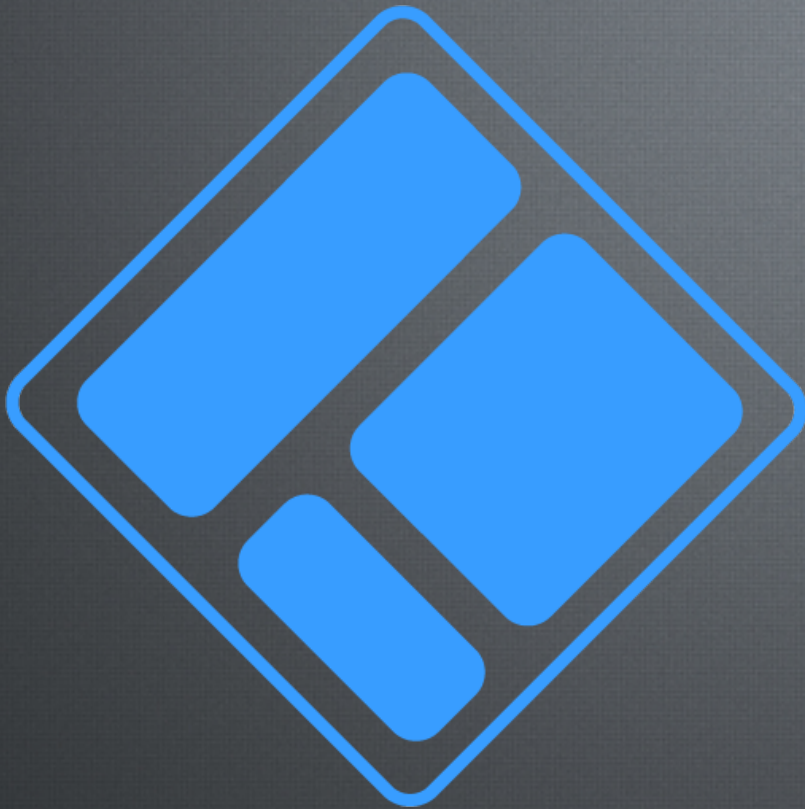


Understanding Angular GPM



+



GPM Structure



Folder Structure

All GlassCat Project Models (*GPMs*) have the same structure. GPM folders contain:

- the `gpm.json` file, which represents the manifest of the GPM
- the `archetype` folder which contains files used to build the project

Pre-install and post-install scripts are located in the optional `scripts` folder.

Expressions

- GPM allows expressions evaluation during build process
- expressions are defined in the form `<% myToken %>`, where `myToken` is evaluated and replaced at build
- all file types specified in the `processedFiles` property list are evaluated
- properties can be passed through command lines

Building The Angular Archetype



To create a new project based on the Angular GPM, you need to call the `glasscat archetype` goal with `angular` as `gpm` parameter:

```
$ glasscat archetype --gpm=[angular] --projectName=[myProject]  
--directory=[myDirectory] --contextRoot=[myContextRoot]
```

The built `angular` project includes:

- Angular 4 dependencies
- Angular Material dependencies

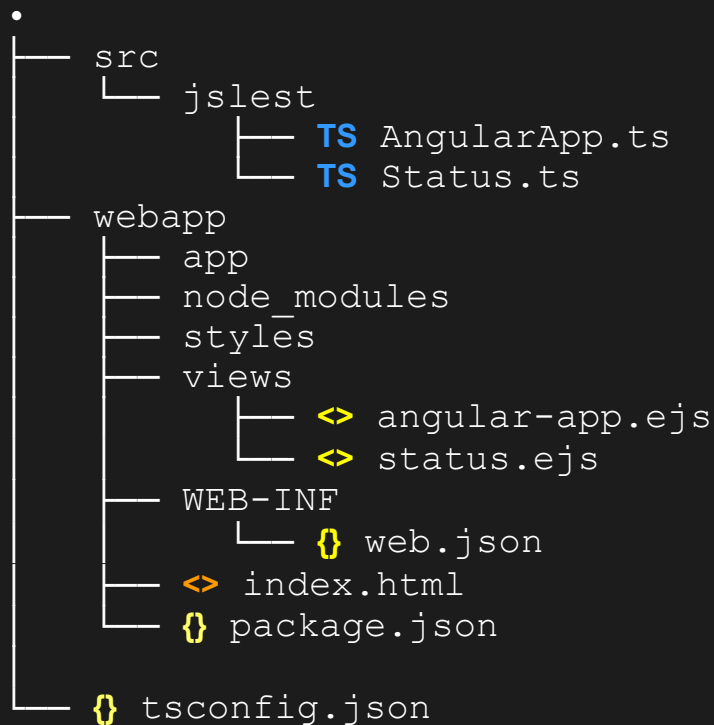
 Angular GPMs use Google Web Fonts as default configuration to load Material icons.

Contrary to Angular-CLI, Angular GPM projects use `SystemJS` to manage dynamic ES modules.

Angular Archetype Structure 1/2



The Angular archetype is designed from a standard JEC Web app:

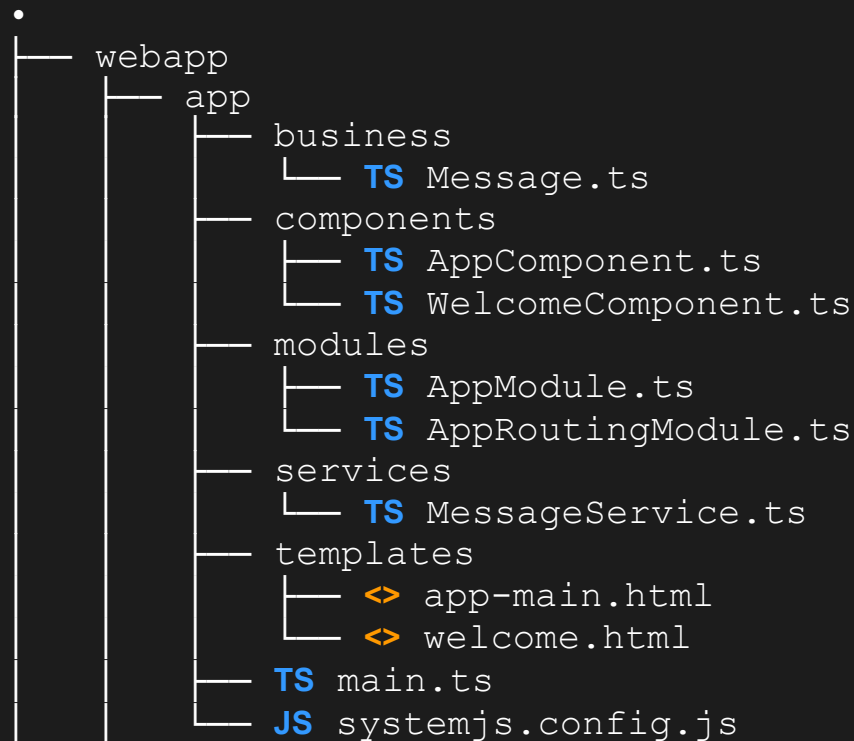


Jslets provides the easiest way to manage cache control capabilities.

Angular Archetype Structure 2/2



The Angular app is designed with functional separation of objects in mind:



Developers can use any code layout to structure their applications.

Angular App Jslet




Angular app is served by using a jslet associated with an EJS template:

```
import { HttpJslet, WebJslet, HttpRequest, HttpResponse } from "jec-exchange";

@WebJslet({
  name: "AngularApp",
  urlPatterns: [
    "/app",
    "/app/",
    "/app/welcome"
  ],
  template: "/views/angular-app.ejs",
})
export class AngularApp AngularApp HttpJslet {

  public doGet(req:HttpRequest, res:HttpResponse, exit:Function):void {
    exit(req, res);
  }
}
```

 You must either declare all routes for the Angular app, or use a generic path (e.g. `app/*`), to ensure accessibility of each URL path.

Securing The Angular App



JEC and jslets allow to easily add security layer to the Angular application:

```
...  
"security": {  
  "roles": [  
    {  
      "name": "ADMIN",  
      "path": "security/AdminRole"  
    }  
  ],  
  "constraints": [  
    {  
      "name": "ConsoleConstraint",  
      "roles": [ "ADMIN" ],  
      "urlPattern": "/app/*",  
      "errorUrl": "/login"  
    }  
  ],  
  "staticResources": [  
    { "urlPattern": "/styles/*" },  
    { "urlPattern": "/node_modules/*" }  
  ]  
}  
...
```

Where to go from here?

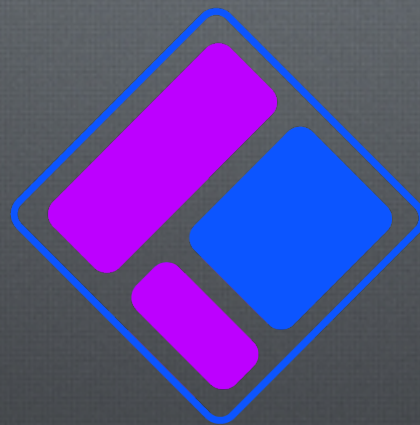


For more information and documentation on GPMs and JEC please visit:

- GlassCat Project
- JEC Sample Projects
- JEC Youtube Channel

The Angular GPM is part of the JEC project:

- JEC project on GitHub



JEC
JavaScript Enterprise Container