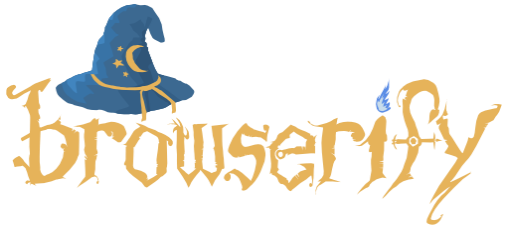
Webpack, Angular overview

Что такое сборка модулей

- Сборка модулей — это просто процесс склеивания группы модулей (и их зависимостей) в один файл (или группу файлов) в правильном порядке.

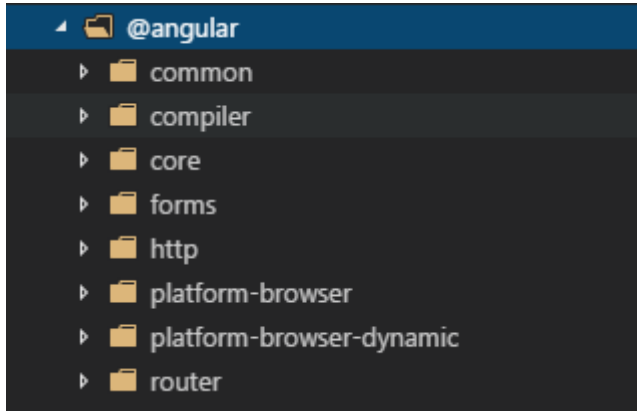
Bundlers



Angular

- Angular is a platform that makes it easy to build applications with the web. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop

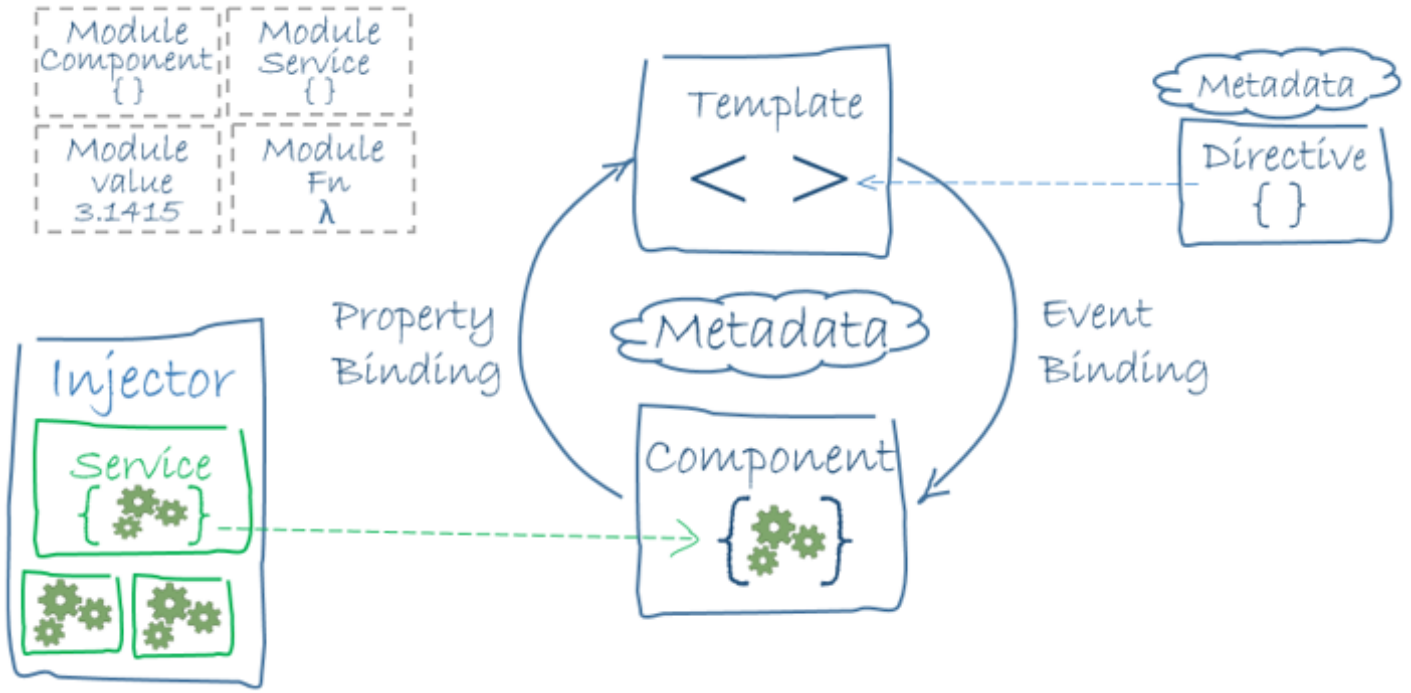
Основные пакеты



- Core
- Compiler
- Common
- Forms
- Http
- Platform Browser
- Platform Browser Dynamic
- Router

Архитектура Angular

- Modules
- Components
- Directives
- Services
- Pipes



Модули

- declarations – классы представления модуля. Существует 3 типа: components, directives, pipes.
- exports – набор деклараций классов которые будут доступны в других модулях.
- imports – набор других модулей, классы которых нужни в текущем модуле.
- providers – дает возможность использовать сервисы глобально в модуле.
- bootstrap – главный компонент (root). Должен устанавливаться в главном модуле.

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Компоненты

- Представление части отображения на странице.
- Взаимодействует с отображением.

```
@Component({
  selector:    'hero-list',
  templateUrl: './hero-list.component.html',
  providers:  [ HeroService ]
})
export class HeroListComponent implements OnInit {
/* . . . */
}
```

Директивы

- Позволяет проводить манипуляции с разметкой.
- Компонент является по сути расширением директивы.
- Существует 2 типа директив: структурные и атрибутивные

```
<li *ngFor="let hero of heroes"></li>
<hero-detail *ngIf="selectedHero"></hero-detail>
```

```
<input [(ngModel)]="hero.name">
```

Сервисы

- Являются обычным классом, декорированным атрибутом @Injectable().
- Могут связывать 2 разных компонента.
- Являются хорошим местом хранения бизнес логики.

```
export class HeroService {
  private heroes: Hero[] = [];

  constructor(
    private backend: BackendService,
    private logger: Logger) { }

  getHeroes() {
    this.backend.getAll(Hero).then( (heroes: Hero[]) => {
      this.logger.log(`Fetched ${heroes.length} heroes.`);
      this.heroes.push(...heroes); // fill cache
    });
    return this.heroes;
  }
}
```

Pipes

- Являются своего рода трансформаторами.
- Изменяют входящий поток данных на желаемый.
- Могут связывать 2 разных компонента.
- Являются хорошим местом хранения бизнес логики.

```
import { Component } from '@angular/core';

@Component({
  selector: 'hero-birthday',
  template: `<p>The hero's birthday is {{ birthday | date }}</p>`
})
export class HeroBirthdayComponent {
  birthday = new Date(1988, 3, 15); // April 15, 1988
}
```

```
@Pipe({name: 'exponentialStrength'})
export class ExponentialStrengthPipe implements PipeTransform {
  transform(value: number, exponent: string): number {
    let exp = parseFloat(exponent);
    return Math.pow(value, isNaN(exp) ? 1 : exp);
  }
}
```

```
@Component({
  selector: 'power-booster',
  template: `
    <h2>Power Booster</h2>
    <p>Super power boost: {{2 | exponentialStrength: 10}}</p>
  `
})
export class PowerBoosterComponent { }
```

Ресурсы

- https://webpack.js.org/
- https://angular.io/

Q&A