

Angular + TypeScript Interview Q&A; Cheat Sheet

Core Angular Questions

Q: What are Angular lifecycle hooks?

A: Angular provides hooks like `ngOnInit`, `ngOnChanges`, `ngOnDestroy` etc. Example:

```
@Component({...})
export class MyComponent implements OnInit, OnDestroy {
  ngOnInit() { console.log('Init'); }
  ngOnDestroy() { console.log('Destroy'); }
}
```

Q: Explain Change Detection strategies.

A: Angular uses Default and `OnPush` strategies. `OnPush` improves performance by checking changes only when `@Input` values change or events fire.

Advanced Angular Topics

Q: How do you use Route Guards?

A: Guards like `CanActivate` and `CanDeactivate` control navigation. Example:

```
@Injectable({providedIn: 'root'})
class AuthGuard implements CanActivate {
  canActivate(): boolean { return !!localStorage.getItem('token'); }
}
```

Q: What are HTTP interceptors?

A: Used for logging, authentication tokens, error handling.

RxJS & Async

Q: Difference between `switchMap`, `mergeMap`, `concatMap`, `exhaustMap`?

A: They control how inner observables are subscribed: - `switchMap`: cancels previous - `mergeMap`: runs concurrently - `concatMap`: queues - `exhaustMap`: ignores new until current completes.

Core TypeScript Questions

Q: Difference between interface and type?

A: Interfaces are extendable, types are more flexible with unions/tuples.

Q: Explain `unknown` vs `any` vs `never`.

A: - `any`: opt-out type checking - `unknown`: safer, must narrow before use - `never`: function that never returns

Generics & Advanced Types

Q: How to use Generics?

A: Example:

```
function identity<T>(arg: T): T {
  return arg;
}
let num = identity<number>(10);
```

Q: Explain utility types like `Partial` and `Omit`.

A: `Partial` makes all props optional. `Omit` removes keys from a type.