

1. What are Services?

Ans: In angular services are reusable classes which can be injected in components when it's needed. Using a separate service keeps components lean and focused on supporting the view, and makes it easy to unit-test components with a mock service.

2. What is the use of @Injectable decorator?

Ans: We are creating an Injectable service using angular @Injectable decorator. When using @Injectable should use parenthesis (). The @Injectable() decorator tells TypeScript to emit metadata about the service.

3. Why \$http Services?

Ans: As the standards are going into next levels these days we are making use of HTTP calls to external API's. To do this angular has built-in HTTP library.

4. How many ways we have for processing this asynchronous operations?

Ans: There are two ways for processing this asynchronous operations:

1. Using Promises
2. Using Observables

5. Explain about XHR and JSONP?

Ans: Most of the browsers supports two Http based requests (XHR and JSONP), and few browsers also supports Fetch XML Http Request (XHR): This request fetches data from server without refreshing page again.

JSONP: This request is mainly used for cross domain requests i.e., calling a service of one domain from another domain.

Fetch: This request fetches with promises.

6. What are the uses of Injectable and Observable in Services?

Ans: Injectable:- which is used to create an injectable service, so that it can be injected anywhere.

Observable:- Used for processing Async operations.

7. Explain about Http, Response, RequestOptions, Headers?

Ans: Http, Response, RequestOptions, Headers: These are imported from angular Http library

- i. Http: for making http request
- ii. Response: The response object which is returned from the request.
- iii. Headers: Headers object can be configured.
- iv. RequestOptions: This holds all the configurations and send it along with the request as an argument.

8. Write a Sample program for services.

```
Ans: import { Injectable } from '@angular/core';
@Injectable()
export class DataService
{
  constructor() {}
}
```

Deccansoft