# JWT definition

Web Token (JWT) is an open standard (RFC 7519) for securely transmitting information between parties as JSON object. It is compact, readable, and digitally signed using a private key/ or a public key pair by the Identity Provider (IdP). So, the integrity and authenticity of the token can be verified by other parties involved. The purpose of using JWT is not to hide data but to ensure the authenticity of the data. JWT is signed and encoded, not encrypted. JWT is a token based stateless authentication mechanism. Since it is a client-side based stateless session, server doesn’t have to completely rely on a datastore(database) to save session information. The big advantage of JWT (Token-based Authentication) is that we store the Token on Client side: Local Storage for Browser, Keychain for IOS and Shared Preferences for Android.

# Data flow

## Backend dataflow

The diagram of the dataflow is presented in the following image.

|  |
| --- |
|  |
| Figure II.1 Dataflow of the backend JWT authentication |

## Frontend data flow of the JWT authentication

The diagram of the dataflow is presented in the following image.

|  |
| --- |
|  |
| Figure II.2 Dataflow of the frontend JWT authentication |

# Http interceptors

Interceptors are a way to do some work for every single HTTP request or response.

Here are a few examples of common use cases for interceptors:

Add a token or some custom HTTP header for all outgoing HTTP requests

* Catch HTTP responses to do some custom formatting (i.e., convert CSV to JSON) before handing the data over to your service/component
* Log all HTTP activity in the console

The method in the interceptor will do the job that we did manually with postman