

Phase 3: Progress from last stage

To begin with, the functionalities that we added since the last stage are the authentication and authorization.

Authentication is about validating an individual's credentials in the application to verify his identity. Authentication determines the elements the system uses to verify one's identity before granting the individual access to anything. An individual's identity is related to what the person knows and it is an important factor to be taken into consideration when it comes to security.

Authorization occurs after an individual's identity is successfully authenticated by the system, which gives access to resources. It verifies user's rights to grant access to resources only after determining the ability to access the system.

In our application we used JWT on the authorization part. JWT can be used for two purposes: authentication and information exchange. *Json Web Tokens* are a good way of securely transmitting information between parties. Because JWTs can be signed, we are sure the senders are who they say they are. The JWT contains a header, a payload and the signature.

The header consists of the type and the signing algorithm being used.

The payload consists of statements about the entity (user) and additional data. We've used the user's Id.

The signature is used to verify the message wasn't changed along the way, and for the token signed with a private key, it is verified that the sender of the JWT is who it says it is.

We also started working on the front end. We started using on the front end side the api methods that we defined in the last stage. For example, we use the GET method for *api/places* in order to get all the available places and display to the user.

```
const url = 'api/users';

class UsersService {
  static getUsers() {
    return new Promise(async (resolve, reject) => {
      try {
        const res = await axios.get(url);
        const data = res.data;
        resolve(data);
      }
      catch(err) {
        reject(err)
      }
    })
  }
}
```

The flow for retrieving the data is the next one: Vue – NodeJs – Laravel. In the client side (Vue) we call the axios.get method for a specific route and in the server side (NodeJs) we get from Laravel the response.

The front end is still work in progress.