```
import { isString } from "lodash";
import {
  AUTH_TOKEN,
  REFRESH_TOKEN,
  EXPIRE_TIME,
  USER_ROLES,
} from "constants/AuthConstant";

export const setAuthenData = (data) => {
  const token = data.tokenData.token;
  localStorage.setItem(AUTH_TOKEN, token);
  const refreshToken = data.tokenData.refreshToken;
  localStorage.setItem(REFRESH_TOKEN, refreshToken);
  const expiredTime = data.tokenData.expires;
  localStorage.setItem(EXPIRE_TIME, expiredTime);
  // const firebaseToken = data.firebaseToken;
  // localStorage.setItem(FIREBASE_TOKEN, firebaseToken);
  localStorage.setItem(USER_ROLES, data.userRoles);
};

export const getExtension = (fileName, indexOf = 0) => {
  let fileExt = "";
  const dotLastIdx = fileName.lastIndexOf(".");
  if (dotLastIdx >= 0) fileExt = fileName.substring(dotLastIdx + indexOf);

  return fileExt;
};
export const makeRandomId = (length) => {
  let result = "";
  const characters =
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
  const charactersLength = characters.length;
  let counter = 0;
  while (counter < length) {
    result += characters.charAt(Math.floor(Math.random() * charactersLength));
    counter += 1;
  }
  return result;
};
export const getNameWithoutExtension = (fileName) => {
  return fileName.substr(0, fileName.lastIndexOf("."));
};

export const formatInputMoney = (n) => {
  if (n <= 0) return 0;
  return n.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",");
};

export const formatMoney = (n) => {
  const price = isString(n) ? +n : n;
  if (price < 0) return 0;
  return price?.toLocaleString("vi-VN", {
    style: "currency",
    currency: "VND",
  });
};
export const isImageURL = (url) => {
  return(url?.match(/\.(jpeg|jpg|gif|png)$/) != null);
}

export const isMediaURL = (url) => {
  return(url?.match(/\.(mp4|mp3|mov|avi|wmv|webm|flv)$/) != null);
}

export const formatterNumber = (val) => {
  if (!val) return 0;
  const formater = new Intl.NumberFormat("vi-VN");
  return formater.format(val);
};

export const parserNumber = (val) => {
```

```
  if (!val) return 0;
  return Number.parseFloat(
    val.replace(/\$\s?|(\.*)/g, "").replace(/(\,{1})/g, ".")
  ).toFixed(5);
};
```