

# ShareJob 設計ver1.2

---

## 実装内容

今回はアカウントの新規登録画面を実装する。以下が機能の概要である

- 新規登録画面にはログイン画面の「新規登録はこちらから」のボタンから遷移できる(これは実装済み)
- 新規登録画面ではユーザー名とパスワードを設定できる
- 入力後、IDが発行され、以降はIDとパスワードでログインできる(IDとユーザー名は別なので注意)
- 今回はPOST通信における暗号化は実装しない

## 処理のフロー

以下が今回の処理のフローである。なお、新規登録画面に遷移してからの記述となる。

1.テキストボックスにユーザー名とパスワードを入力する 2.ユーザー名とパスワードをPOSTする 3.バックエンド側でユーザー名とタイムスタンプによるIDを発行する(万が一にもIDが他と被らないための対策) 4.DBにユーザー情報を登録 5.レスポンスでIDを返す 6.IDとパスワードを表示し次回からこのIDとパスワードでログインするように促す 7.同時に戻るボタンを用意しておいてトップ画面に戻れるようにしておく

## 詳細設計

以下に詳細設計を示す。なお以下に示す場合以外に変更する必要が出てきた場合は臨機応変に変更しても構わないが変更した旨は連絡すること。(本設計書にも記載してくれると尚よし)

### バックエンド側

- controller
  - SignUpController ユーザー名とパスワードをPOSTで受け取ってユーザー名とタイムスタンプでIDを生成し、ユーザー名、ID、パスワードを serviceに投げて登録処理を行う。その後IDをレスポンスとして返す。
- service
  - UserService UserServiceImplのメソッド定義を行う
  - impl
    - UserServiceImpl ユーザー名、ID,パスワードを元にユーザー登録をUserRepositoryで行う。(JPAではユーザー追加のメソッドは実装済みなので今回はrepositoryの実装はいらない)
- form
  - SignUpForm POSTする情報を定義する

### フロントエンド側

- views
  - SignUp.vue ユーザー名とパスワードを受け取ってPOSTしたのち、レスポンスを受け取って ShowLoginInfo.vueに画面遷移する
  - ShowLoginInfo.vue IDとパスワードを表示し、次回からこのIDとパスワードでログインすることを指示する。その後TopView.vueに遷移する
  - PageHeader.vue ShowLoginInfo.vueを表示している際のTopView.vueへの遷移ボタンを追加する

- router
  - index.js 今回追加したviewsの追加
- store 状況に応じてページ間での情報を保持したい際に変更が必要な可能性があり

## 備考

### ID生成方法

IDはユーザ名+タイムスタンプを一方方向性ハッシュ化を用いて暗号化したものとする。(ハッシュ化した際の特許文字は除く)

Java 標準ライブラリで暗号技術を使う <https://qiita.com/Qui/items/6aa46d428e5bfe9b8d37#>